Instantly Available Power Managed Desktop PC

Design Guide

Revision 1.2
9/25/98

Intel Corporation

DISCLAIMER

The information contained in this design guide represents the current view of Intel Corporation on the issues discussed as of the date of publication. Because Intel must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Intel. Intel does not guarantee the accuracy of any information presented in this design guide. This design guide is for informational purposes only and is provided "AS IS." INTEL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter described in this document. The furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property rights, either express or implied, by estoppel or otherwise.

Intel does not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Intel disclaims all express and implied warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Intel does not make any warranty of any kind that any item developed based on any proposal, specification, sample or example will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Intel shall not be liable for any damages arising out of or in connection with the use of this design guide, including liability for lost profit, business interruption, or any other damages whatsoever.

Intel and Pentium are registered trademarks and MMX is a trademark of Intel Corporation.

Third-party brands and names are the property of their respective owners.

Revision table

Date	Revision	Changes
5/98	1.1	Added description of 3.3Vaux ECR. Changes made to power delivery section to incorporate minimum subsystem requirements
9/98	1.2	Sections changed and modified 2.31, 2.4.1, 2.11 Sections added: 2.3.2.2 – RDRAM support 2.12 – Keyboard and mouse wakeup considerations 2.13 – S4, S5 wakeup support 2.14 – 440BX design checklist 5.4, 5.5 – Power distribution options, cost, RDRAM support 6. – Software considerations

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The PC and the Internet have become an integral part of every day life in the home, office and on the road. The ability to readily access the latest, up-to-the-minute information has become a baseline user expectation. Meeting these expectations requires that the next generation of PC be instantly available to both the user and to communications applications. At the same time the PC must become more efficient in its use of electricity as more computers come on-line with a limited supply of inexpensive energy.

The Environmental Protection Agency's Energy Star office-equipment labeling program has motivated PC manufacturers to pay closer attention to the energy consumption of their products while in the "Standby" state, however with the current BIOS-based power management techniques power savings typically come at the expense of connectivity and/or performance. Intel's 1998 Desktop Power Management Initiative, The Instantly Available Power Managed PC brings together many new technology ingredients that enable these user expectations to be met for the volume PC market segment by the second half of 1998. The Instantly Available PC architecture ensures that the PC gets all the power it requires for full performance when being used, yet, aggressively power manages devices that are not in use to consume less power.

The most challenging technical hurdle to overcome is in achieving extremely low power consumption while maintaining a system's LAN or modem connectivity on a desktop PC/workstation. The overall goal of having the PC be instantly available while also maximizing energy savings will require the designer to make implementation tradeoffs.

The reader is referred to Intel's Power Management Web page (http://developer.intel.com/ial/powermgm/) for roadmap updates, power management white papers, and other related information.

1.1. Objective

This design guide describes the platform integration of new power management technologies that deliver on the user expectation of instant availability regardless of the system's power management state.

This document's objective is to provide the reader sufficient information detailing the integration of the hardware, and software technology ingredients required to build Instantly Available desktop PCs. This "how to" guide is intended to equip the PC development community with the information that will help facilitate the broad adoption, and ease the integration of these new capabilities into all future desktop PCs.

Note that the integration of the technologies described in this guide delivers a baseline set of platform power management capabilities that can be applied to ALL classes of Intel Architecture PCs including, besides desktop PCs, mobile computers, workstations and servers.

1.2. Target Audience

The target audience for the Instantly Available PC design guide includes peripheral component and subsystems designers, PC systems designers, PC systems integrators, and software developers. The design guide is technical in nature, specifically targeting engineers.

1.3. Target Time Frame for Volume Market Segment Penetration

The technology ingredients necessary to deliver the Instantly Available PC in volume, including home and business operating system support, are expected to be in position by Q3 of 1998.

1.4. Related Documents

- ACPI, Advanced Configuration and Power Interface Specification Revision 1.0, http://www.teloport.com/~acpi
- ACPI Implementers Guide (Draft) 4/4/97 Intel/Microsoft/Toshiba
- 1394 TRADE ASSOCIATION, Power Specification Draft .09
- Universal Serial Bus Specification, Revision. 1.0, http://www.usb.org
- PCI Bus PM Interface Specification, Revision 1.0, http://www.pcisig.com
- Instantly Available PC Power Delivery Requirements and Recommendations, http://developer.intel.com/solutions
- OnNow White papers, http://www.microsoft.com/hwdev/onnow.htm /
- Device-Class PM Specifications, http://www.microsoft.com/hwdev/ONNOW.HTM#pmSPECS
- PCI SIG Power Management Working Group's 3.3Vaux ECR.
- TIA-695 voice modem specification, available from Global Engineering: (800) 854-7179
- ITU V.voice specification, http://www.itu.ch/itudoc/itu-t/rec/v.html
- Rambus RDRAM specifications, http://www.rambus.com

1.5. OS Directed Power Management Overview

Operating System Directed Power Management (OSPM) is a model for Power (and system) Management in which the operating system (OS) plays the central role using global information only it possesses to optimize the system's behavior. The key advantage of OSPM over current state of the art BIOS-based power management is that it offers system-wide visibility where all elements of the system operate as an integrated, power-managed whole according to defined roles, responsibilities and standard interfaces. The standards-based method by which the OS, hardware and applications communicate along with the OS's knowledge of application workloads within the PC enables robust platform power management heretofore unachievable by existing Advanced Power Management (APM) based systems.

The centerpiece of OSPM is the Advanced Configuration and Power Interface Specification (ACPI). ACPI provides a standard set of interfaces for hardware and applications to communicate power control capabilities and requirements to the operating system. An ACPI enabled operating system binds all power managed elements in the system to establish an intelligent platform-wide power management solution.

For a detailed description of ACPI specifications, refer to *ACPI, Advanced Configuration and Power Interface Specification (Intel/Microsoft/Toshiba) Rev. 1.0.*

1.6. Benefits

The end user benefits of the Instantly Available PC manifest themselves differently depending upon the PC's usage.

1.6.1. Office Desktop PC

Current state of the art office PCs integrate manageability capabilities intended to reduce their Total Cost of Ownership (TCO). Being able to manage a PC's software inventory from across the network is one example of how the cost of ownership can be reduced. Today's manageable PCs have a special mode where they can be remotely rebooted to a manageable state from which, for example, new software can be installed, diagnostics run, etc.

The Instantly Available PC architecture takes these "Wake on LAN" manageability capabilities to the next level while keeping the "off" energy consumption at the same low levels. Architectural improvements include:

- replacing the boot process with a resume process
- keeping the PC completely connected to the LAN

Rather than rebooting the PC, network traffic (or programmed event) of interest cause the PC to resume to precisely the same system state it was in before entering a sleeping state. Intelligent LAN pattern filtering allows connection to the network to remain active for programmed access out onto the net as well as for incoming network traffic enabling file sharing, print server, and web server applications to function even when the PC is "off".

1.6.2. The Home PC

The home PC, whether a traditional desktop multimedia PC or a high end PC in the family room, benefits by taking on many of the attributes of consumer electronics devices already present in the home. The PC appears to be off in that there is no noise due to fans or disk drives, yet it can snap back to its fully ready state within seconds of the push of a button, or respond to the phone ringing in time to fully service the

call for voice, fax or answering machine applications. Architectural improvements defined for the modem enable the PC to fully resume prior to the second ring, along with new capabilities that also enable the capture and retrieval of otherwise lost Caller-ID information. The user has become accustomed to this type of instant availability from existing consumer devices such as the television and the telephone.

2. ACPI Motherboard Design Considerations

This chapter describes design issues that the designer is expected to encounter while implementing an Instantly Available PC. Figure 1 below is a conceptual drawing of the system's motherboard. All of the main system components are shown. The following subsections describe key capabilities of the main motherboard components while also highlighting design issues that need to be considered.

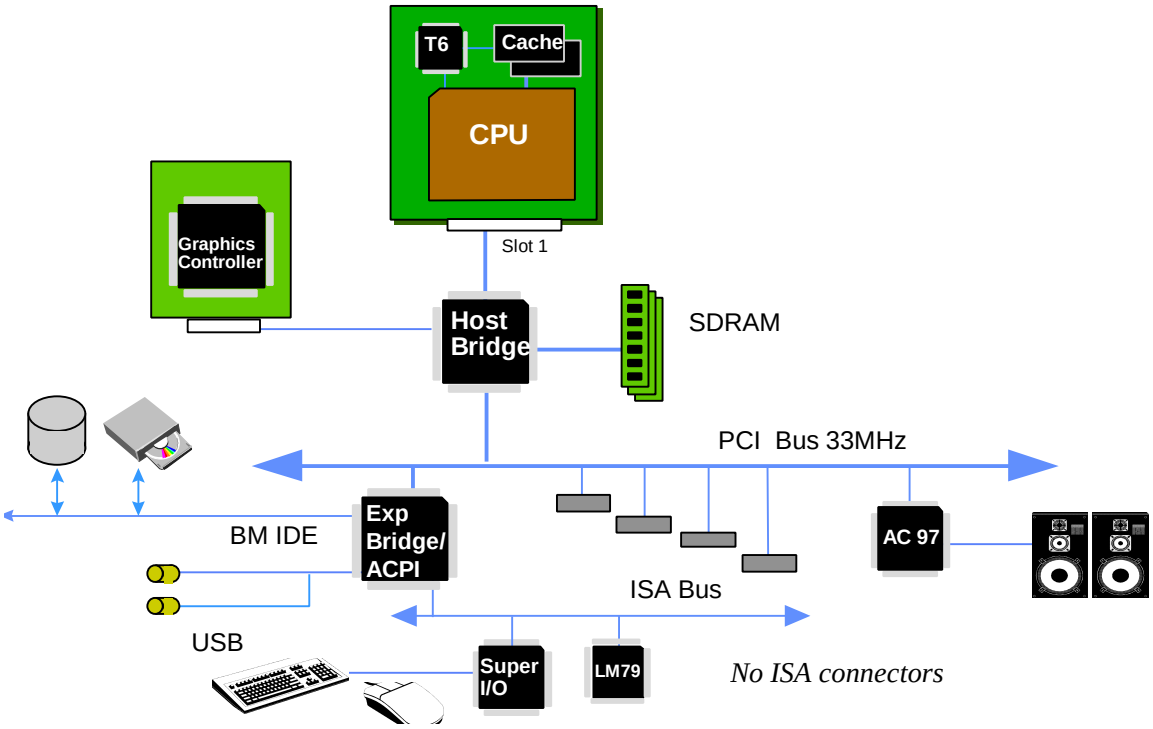

Figure 1. Motherboard Block Diagram

2.1. Sleep State Support

The motherboard must be designed to support system states as defined by the ACPI specification. In an ACPI platform there is the working state, G0, and a number of available "sleep" states to choose from when implementing the system. S1, S2, S3, S4 and S5 are all sleep states as defined by ACPI. A system designer is free to implement all of the sleep states, however during normal operation the system will toggle between the working state and typically only one or two supported sleep states.

Since typically only one or two sleep states will ever be used by the OS, the choice of which sleep state(s) to implement in support of the Instantly Available PC must balance both the functional requirements as well as the implementation cost. From a functional perspective S4 (Save to Disk) and S5 (Soft Off) sleep states impose resume latencies that are too lengthy to support "Off yet Communicating" capabilities and so they cannot be used if these capabilities are desired. S5 could be implemented for just what its name indicates, a soft off, where minimum power is consumed, only powering the "on" button. S4 adds no real value to the Instantly Available PC other then to provide a fail-safe resume vehicle, which could be used if, for example, a power failure had occurred while the system was sleeping in S3.

2.1.1. Recommended Sleep States

The ideal sleep state for the Instantly Available PC is S3 (Suspend to RAM). In this state over 80% of the PC's power is shut off enabling a breakthrough in power savings. The resume time from S3 (from DRAM) is targeted to be less than 5 seconds which also enables "Off yet Communicating" capabilities. The implementation of S3 relies on the presence of an auxiliary power source as well as a split power plane system design. These two key elements enable the system to power, independently, any communications "keep alive" logic while the majority of the PC is unpowered.

To summarize, platform support for the S3 sleep state, split power planes, and a dual mode power delivery (which will be described later), make up the nucleus of an Instantly Available power managed PC.

Since the number of wake devices is likely to grow over time while the auxiliary supply that will be found in most dual mode power delivery subsystems will have limited capacity (for cost reasons), the S1 state should also be implemented as an alternative sleep state to S3. If for example the power management policy manager were to determine that there are too many functions enabled for wake up, i.e. more than can be adequately supported by the auxiliary power source, then it could decide not to enter S3, but rather S1.

There are a number of different ways that a system could be designed to handle the budgeting of auxiliary power as it relates to power management policy. This is an area where the system designer has to make tradeoffs, which will ultimately dictate how he intends to differentiate his product.

2.1.2. S2 Design Considerations

S2 is defined as a sleep state where the processor's power may be removed. Platform implementations of S2 may allow the processor and PCI bus clocks to be stopped upon entering this state.

System clocks are restarted when leaving S2, in transition back to the working state. The clocks are stopped to achieve the maximum power savings while all elements of the system, with the exception of the processor, are still fully powered. Careful system analysis needs to be done to fully understand all of the issues associated with stopping and restarting the clocks.

S2 implementation is not recommended for the Instantly Available PC since its incremental power savings over S1 does not appear to justify the added design complexity associated with its implementation.

2.2. Processor

The ACPI specification defines the C0, C1, C2 and C3 processor power management states. The processor states are briefly defined in the table below:

State	Key Characteristic	Implementation Details
C0	working state	Processor is running cycles
C1	processor is stopped	Halt instruction is executed.
C2	processor is stopped	Stop clock and stop grant signals are asserted.
C3	processor is stopped	Lower power state than C2. Cache context is maintained. (However the processor does accept any snoop requests, which assumes no other memory traffic while in C3). The power savings over C2 may not justify implementation of C3 in the desktop environment.

Table 1. ACPI Processor Power Management States

The support for these states is included in Intel's processors and core logic chipsets. Therefore there is nothing specific that the designer needs to implement in order to support the processor states for the Instantly Available PC.

Since the supported processor power management states are communicated to the OS at boot time by the BIOS, the system designer has the option of choosing which states to report and support in his system.

2.3. Bridge Architecture with ACPI support

2.3.1. The Host Bridge

The Host Bridge is responsible for bridging host transactions to either system memory, the PCI Bus, or the AGP graphics port. It also bridges AGP and PCI mastered transactions to memory while initiating the appropriate host processor cache snoop cycles.

Since the memory controller is a part of the host bridge the designer needs to determine if the power to the host bridge is applied during the S3 state.

The simplest way is to apply the power to the host bridge while the system is in suspend to RAM state. This method guarantees that all the memory control registers are maintained and that there will not be a inappropriate signaling sequence on the memory control lines that could cause the memory array to exit self refresh state. This approach has been used on Intel reference platform.

The other approach is to remove the power from the memory controller when the system transitions to S3 state. In this case first all the registers in the host bridge need to be saved in non volatile memory. Next the proper signaling sequence need to be generated by the memory controller to put the memory devices in self refresh state. Finally, the control lines between the memory controller and the memory devices need to be separated and driven to the known valid state to prevent undesired exiting from self refresh state.

The reverse sequence needs to be applied when the system wakes up from S3 state.

The host bridge also needs to communicate with the ACPI controller which resides in the expansion bus bridge. The host bridge tracks system power states and generate proper control signals for the system. Suspend status control signals are used by Intel core logic chipsets to synchronize state machines between the core logic components. The ACPI controller, for example, uses the suspend status control signals to command the memory controller to signal the Suspend to RAM entry sequence when transitioning the system to the S3 state.

2.3.2 System Main Memory

Considering that Suspend to RAM (STR) is the cornerstone of the Instantly Available PC it is important that the chosen memory technology supports it well.

Synchronous DRAM (SDRAM) is recommended over conventional fast page or EDO DRAM. Synchronous DRAM supports a self refresh mode in which the power for the DRAM is significantly reduced. For example the typical DIMM module may use up to 3 W (1A) when in an active state, however it only uses between 50-100 mW when put in the self refresh mode. Therefore with SDRAMs even very large memory configurations will be dissipating less than 0.5 W.

Additionally, SDRAMs do not need to have their clock inputs running while in the self refresh mode. This simplifies the memory controller design while also reducing the memory controller's power consumption during the sleep state.

As indicated in the previous section, the memory controller is responsible for generating the DRAM signaling sequence, which starts and stops the self refresh mode. Because of this requirement, the memory controller has to be part of the same power plane as the DRAM components as shown in the power plane diagram, Figure 3.

2.3.2.1 System Main memory - RDRAM technology.

The RDRAM memory technology is considered as the next generation memory technology for the future system designs.

The RDRAM devices support power management and have various modes of operation. The following modes of operation are typical to the RDRAM devices: active read, active write, active idle, nap and power down. The power down mode is the mode in which the RDRAM devices are placed when the system enters S3 state. The typical power currently published specifies that each device will consume a maximum of 1.5 mA.

The RDRAM power requirements in the power down mode have been given as an example in Table 2 below. It is assumed that the 64 Mbit devices will be used and that the typical current per device in power down mode will be less than 1.5 mA.

Rdram mode	Icc (mA)/ 64 Mbit device	Total memory (MB)	number of devices	Max power (W)	Notes
Powerdown (S1, S3)	1.5	32	4	0.015	64 Mbit devices
Powerdown (S1, S3)	1.5	64	8	0.030	64 Mbit devices
Powerdown (S1, S3)	1.5	128	16	0.060	64 Mbit devices
Powerdown (S1, S3)	1.5	256	32	0.120	64 Mbit devices

Table 2. Max RDRAM power in S3 state

The RDRAM power that needs to be budgeted in the S3 state can be approximated based on the Table 2.

For example for 256 MB of memory the max standby power is only 120 mW. (48 mA of current)

2.3.3. The Expansion Bus Bridge

The Expansion Bus Bridge is responsible for bridging expansion buses such as ISA, USB and IDE to the PCI bus, and ultimately to system memory.

2.3.3.1. The ACPI Controller

Intel core logic-based platforms integrate the ACPI controller in the Expansion Bus Bridge.

The optimum ACPI-enabled core logic chipset should provide a system designer with the ability to get the desired result while also enabling him to differentiate his products from those of the competition.

For implementation flexibility the ACPI-enabled core logic chipset should support power management features that can be easily mapped to the ACPI defined power/sleep states.

Intel ACPI core logic chipsets provide the following features in support of the ACPI specification:

- Power on Suspend
- Suspend to RAM
- Suspend to disk

It is up to the system designer to map the ACPI controller's supported power management features to defined ACPI sleep states. Typically, Power On Suspend would be used to implement either the S1 or S2 state. The Suspend to Ram feature maps directly to ACPI's S3 state.

The ACPI controller implements sleep state machines to control system operation. The sleep states are entered by accessing the sleep type registers of the ACPI controller. Upon writing to a sleep type register the appropriate state machine logic executes the necessary sequence of the events, ultimately leaving the system in the desired sleep state.

The ACPI controller also generates signals to control the power planes as well as to turn the power supply mains on and off.

2.3.3.2. IDE Controller

The standard IDE controller is built into the Expansion Bus Bridge. The controller is capable of turning the devices On and Off when they are not in use. The IDE controller typically supports ON and OFF states that can be mapped into ACPI device states D0 and D3. The other states such as D1 and D2 are not supported

2.3.3.3. USB Controller

A USB controller is also integrated into the Expansion Bus Bridge. Note that in order for USB peripherals to be able to wake the system from the S3 state, the USB controller must derive at least some of its power from the auxiliary power source. Without auxiliary power the USB controller would be unable to respond to a USB wake event (from, for example, a USB modem) when main power has been turned off. For detailed information on USB's power management capabilities please refer to the USB Revision 1.0 Specification, available at http://www.usb.org.

2.4. PCI Add-in Connectors

The PCI SIG has specified two new, formerly reserved, connector pins that are used in support of the Instantly Available PC. The new pins are PME# and 3.3Vaux. The PME# (Power Management Event) pin is used to wake the system in response to a PCI power management event such as the phone ringing. The second pin, 3.3Vaux, defines the standard for delivery and consumption of auxiliary power for PCI add-in cards. The addition of these two pins enables split power plane add-in designs that can be configured to wake the system from the S3 state.

Additional information is available in section 3 of this design guide which focuses on PCI bus "Off yet Communicating" design considerations.

2.4.1. Connecting PME# and Vaux signals

PME# is connected to the ACPI controller via a dedicated input pin. The PME# motherboard signal is a wired 'OR' connection of all PME# output drivers in the system. The input pin and related ACPI controller logic must be powered by auxiliary power such that it remains active regardless of the system's power management state.

Intel's Instantly Available PC Reference Platform connects the PME# signal network to a general purpose event pin - ***GPI1*** on the Expansion Bus Bridge. This pin is capable of monitoring PME# even when the main power has been removed. It is important to notice that the PME# pin on the ACPI controller is a dedicated pin just for PCI devices. All the devices sharing PME# pin need to be complaint with the PCI PM 1.0 or later specification.

The PME# signal is connected to pin A19# on all the PCI connectors.

Vaux signal delivers auxiliary power to the wakeup devices in the system. The Vaux voltage is defined as 3.3V signal.

Vaux signal is connected to the Vaux pin of PCI connectors (pin A14) is being driven by the dual voltage generation circuit. The Vaux signal should be routed as the wide trace on the motherboard ensuring that the voltage at the input to the PCI device is within specification. (3.3V +/- 10%).

2.5. ISA Add-in Connectors

ISA add-in adapters were not designed to support power management features and are not recommended. The system designer is encouraged to design an ISA ***slot-less*** system.

Note that a system can be designed to support the S3 state even if ISA add-in cards are installed (provided they are ISA Plug and Play compliant). In this case the ISA bus and its peripherals would experience what they would perceive to be a cold power on reset when main power is restored. Unfortunately however, while "Off yet Communicating" capabilities are certainly not supported by any ISA subsystems, they may also be broken for the rest of the system with ISA add-in cards installed due to a greatly increased resume latency associated with the time it would take to rediscover, re-enumerate, and re-activate them upon wake up.

2.6. Thermal/Noise Control

2.6.1. Thermal Sensors

Thermal sensors could be implemented on the motherboard to monitor CPU and system ambient temperature. The purpose of the sensors is to control fan noise during normal system operation. Typically one or two system sensors are sufficient.

The temperature reading is used as the input to the ACPI thermal policy. Based on the system temperature the required speed of the system fans is determined. Note that the intention is to stop the fans from spinning while an Instantly Available PC is in an "off yet communicating" sleep state (i.e. S3)

2.6.2. Fan Speed Control

Programmable fan speed control enables an Instantly Available PC to manage its emitted acoustic noise levels while in the working state (temperature permitting).

The most cost sensitive implementations of Instantly Available PCs may choose to not implement any programmable speed control at all. In these systems the fans are powered by the Vmain power source such that when the system enters the S3 state and the Vmain supplies turn off, the fans turn off as well. This approach may also be well suited for a business environment where the fan is either full on or full off. PCs that are intended to be deployed as home entertainment centers however, may demand a more active acoustic noise management scheme to reduce noise while the PC is playing music for example. It is for this "consumer-class" of Instantly Available PC that programmable fan speed control is intended.

Two commonly used schemes for controlling fan speed are:

- Linear voltage control
- Pulse width modulation (PWM)

Pulse width modulation allows more flexibility for the system designer and allows finer fan speed granularity. Fan speed control is recommended for both fans in the system:

- power supply fan
- system fan

Note that some systems may also have a CPU fan. For these systems the CPU fan speed could also be controlled by the ACPI thermal policy manager.

The control logic should support 3 spinning speeds as well as the off state.

High speed is intended to be used when targeting maximum system performance

Medium speed should be used for the nominal/normal system operation.

Low fan speed could be used when the system is in the low performance mode or is in the idle state.

The system designer must ensure that when the fan speed changes the noise introduced as a result of that change is not noticeable to the user, i.e. it happens gradually. This consideration may lead the designer to implement additional fan speeds.

When the system enters the S3 state both system fans should be turned off. The ability to adequately cool the system without fans while in the "off yet communicating" S3 state is the key to realizing a completely quiet PC when "off".

2.7. Thermal policy. Active versus passive cooling

Fans are controlled based on the thermal policy implemented in the operating system.

The thermal policy is based on trip points, which the OS uses to turn on/off fans (active cooling), throttle the CPU (passive cooling) and shut down the system (critical cooling). Two sets of trip points are required since the user/OS has the ability to change the cooling policy (quiet or performance).

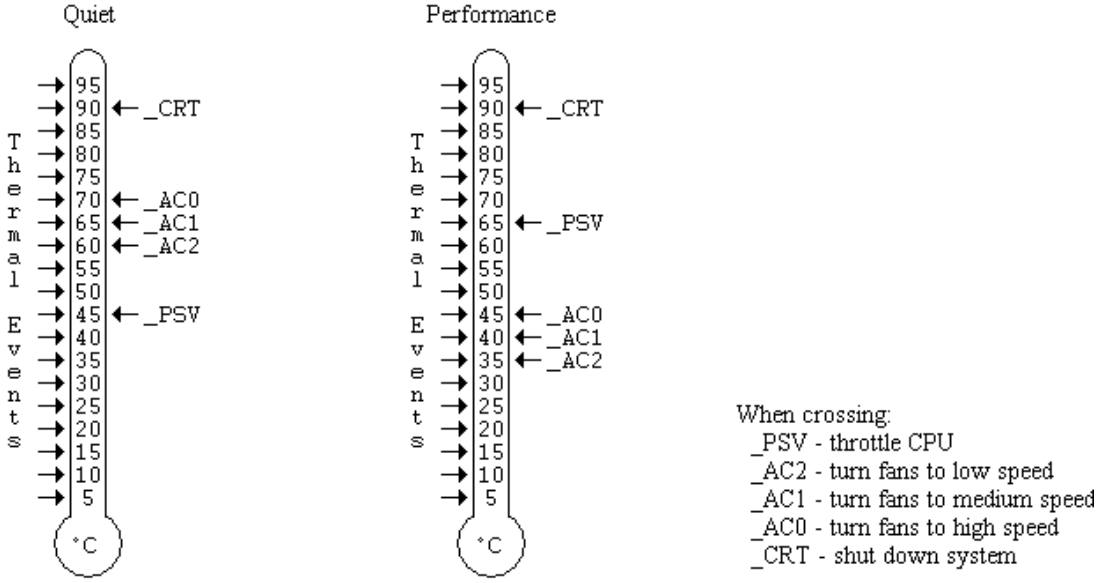

Figure 2. Thermal Zones

The _WAK control method:

Changes the external temperature thresholds to _CRT
Changes the internal temperature thresholds to the current temp ±5 °C
Notifies the OS to re-evaluate the policy temperatures.

The left side of Figure 2 above illustrates ACPI's passive thermal policy. If, for example the system was configured for the quietest operation possible it would start to throttle CPU performance down when the temperature reached a certain level. All of the fans are still stopped at this point. If the system temperature continues to rise the fan speed is throttled upward.

The active thermal policy (on the right side of Figure 2) is intended to achieve the highest system performance. This policy mandates doing as much thermal and noise management as possible by first throttling up the fan speed before having to throttle the performance of the CPU.

In a well designed system utilizing the active thermal policy, the system should never be required to throttle back processor performance. The high fan speed setting should be sufficient to cool the system while running its maximum work load.

Thermal policy can be configured to either generate a System Control Interrupt (SCI) based on the temperature crossing certain thresholds, or based upon any 5 °C temperature changes.

When there are multiple fans in the system the system designer should model the air flow throughout the system to determine whether both fans can be controlled by the same fan speed control, or whether they should be designed to operate on separate fan speed controls (i.e. system designed to run with independent fan speed settings).

2.8. Power Distribution and Control

2.8.1. Power plane control

The figure below shows the power plane partitioning of an Instantly Available PC. Three new voltages have been introduced: 3.3V dual, 5V dual and 3.3V standby. The concept of dual mode outputs is described in Section 5 of this design guide.

The memory subsystem's power plane is connected to the 3.3V dual output of the power supply. The same output is also connected to the PCI connectors 3.3Vaux pin in support of add-in PCI wakeup devices.

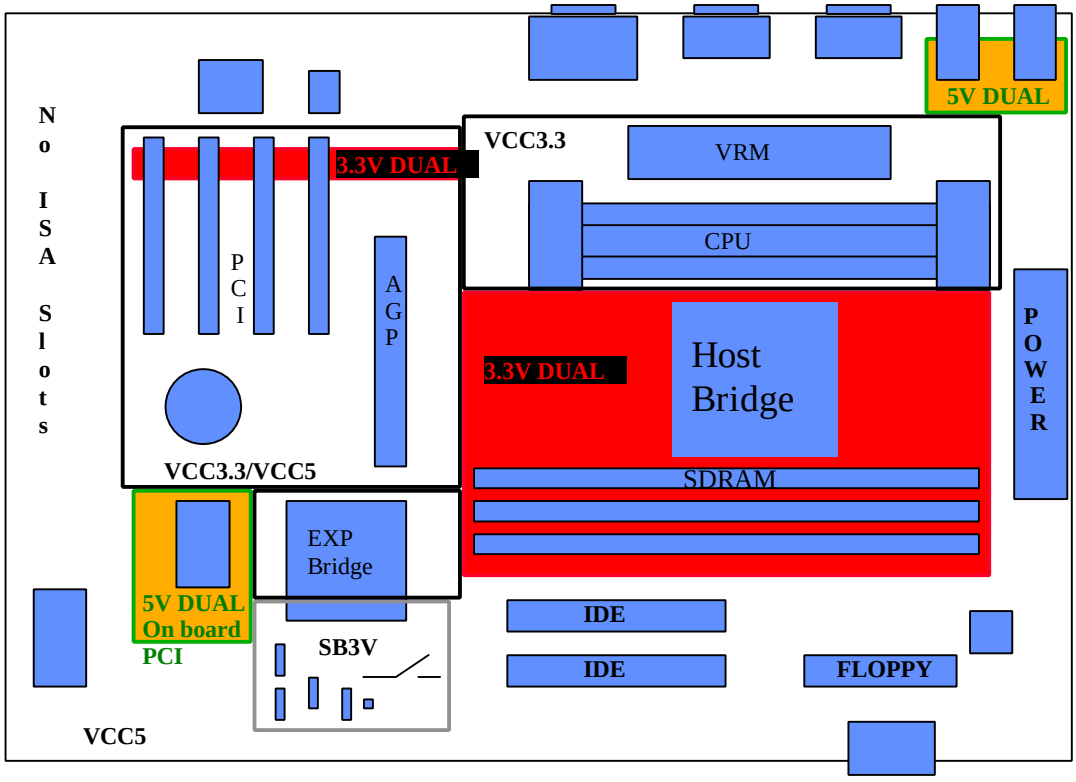

Figure 3. Motherboard Power Plane Partitioning

The following table shows a typical mapping of the motherboard components to the Instantly Available PC various voltages.

Voltage planes	Consumers
3.3V	3.3 V PCI, 3.3 V components, processor cache, Expansion Bus Bridge
5V	processor VRM, PCI and other 5 V components on the motherboard
3.3V dual	Host bridge with memory controller, SDRAM components, memory clock driver, 3.3Vaux PCI connector auxiliary power pin
5V dual	USB devices (USB wakeup capabilities will be included in the future devices), 5V based PCI motherboard devices intended to be wakeup devices.
3.3V always present standby	This voltage is always present when the system is plugged in to the AC socket. It is intended to be connected only to the ACPI controller's resume well, the power button and power indicator (LED).

Table 3. Split Power Plane Voltage Distribution

The remainder of the platform voltages are what today's desktop power supplies already comprehend.

2.9. Suspend/Resume, Power Button

The power button functionality has been redefined for ACPI systems. The power button now acts as a sleep button that does not actually turn the machine on or off. Instead it signals the OS to either wake the machine up or put the machine in the sleeping state.

If the system is in the working state when the user presses the sleep button, the button push will signal the OS to transition to the sleeping state that had been specified by the user. If, on the other hand, the system is in the sleeping or soft off state when the user presses the button the system wakes up and signals the OS to transition to the working state.

In an emergency situation (for example, the system software is locked up), the user can press the button for 4 seconds to force the system to go directly to the S5 state. The user will most likely lose his working data, but this is used only in an emergency.

ACPI defines two types of these "soft" buttons: one for putting the machine to sleep and one for putting the machine in the S5 "Soft off" state. This gives the OEM two different ways to implement machines: A single button model or a dual button model.

In a single button model, the button can be used as a power button or a sleep button as dictated by user settings. A two-button model has an easily accessible sleep button as well as a separate power button. In either model, an override feature that forces the machine off or to perform a cold reset without OS consent is also required to support various atypical, yet problematic situations.

First generation desktops will have fixed, 4 second power/restore override button functionality. In other words, if the user presses the button for more than 4 seconds the soft suspend/resume signaling will be overridden and the button will either force a cold system reset or the soft off transition (depending upon the power management state of the PC when the button is initially depressed)

2.10. Power button implementation

The Instantly Available PC reference platform implements the power button as a fixed power button, so no ASL code has to be written to directly support the power button requirement. The reference platform's ACPI-enabled chipset integrates the fixed power button logic as shown in section 4.7.2.2.1.1 of the ACPI Specification.

2.11. System power state indicators

The power LED must, as per the ACPI specification, indicate the power management state of the machine to the user. (i.e. whether the system is in a working or sleeping state). The power LED must be present on all ACPI compliant PCs.

Several options are available. Two LEDs could be used to indicate the working versus sleeping status. Dual (multi-color) LED could also be used for this. If the system implements voice communications a separate blinking LED is recommended for "voice message pending" indication.

System state	Message waiting	LED
G0/S0	No	Green
G0/S0	Yes	Green - blinking
G1/S1/S3	No	Orange
G1/S1-S3	Yes	Orange - blinking
G1/S4	N/A	OFF
G2/S5, G3	N/A	OFF

Table 4. One LED system indicators

System state	Message	Power/Normal LED	Sleep LED	Message Waiting LED
G0/S0	No	Green	OFF	OFF
G0/S0	Yes	Green	OFF	On or blinking
G1/S1/S3	No	OFF	Orange or blue	OFF
G1/S1-S3	Yes	OFF	Orange or blue	On or blinking
G1/S4	N/A	OFF	OFF	OFF
G2/S5, G3	N/A	OFF	OFF	OFF

Table 5. Multiple LED system indicators

2.12. Keyboard, mouse and other non standard device wakeup considerations

In typical ACPI system the legacy keyboard and mouse wakeup are not recommended. The users are encouraged to use the power buttons, which now causes the system to enter the sleep state instead of the

OFF state. This initiative is trying to break “screen saver” paradigm in which the user can use keyboard or mouse to exit the screen saver application.

Please note that the IRQ events are not serviced while the system is in the sleep state.

However the ACPI specification allows implementing keyboard and mouse wakeup features. To support such wakeup events the keyboard, mouse and the keyboard controller has to be powered while the system is in S3 state. The BIOS control method has to be written to let the OS know who cause the wakeup and how to service the event.

To simplify the designs it is possible to design the circuit, which based on the pressing the keyboard key generates an impulse. This circuit can be connected in parallel with the power button.

Please note that if the controller is not compliant with the PCI PM specification it can not use the PME# pin for the wakeup event.

2.13. S4 and S5 state wakeup considerations

The ACPI specification describes S4 and S5 states as the states where all platform power (except the battery) has been removed from the platform. However there maybe some application in which the system designer decides to keep some circuits powered while in S4 or S5 state.

An example is a remote power up when the new system needs to be configured and the OS needs to be installed. For this application leaving power to the network adapter enables system administrator to generate wakeup event and install the software over the network connection.

In the case described above the sleep state are not defined since the operating system is not yet installed.

Please note that wakeup from S4 state is not recommended due to the long resume delay. The wakeup from S5 state is not supported by the operating system. All context is lost in S5 state and wakeup devices are disabled. Power button is the only event that can wakeup the system from S5 state.

2.14. 440BX system design checklist

This section has been added to highlight several design issues encountered while implementing Instantly Available reference platform.

Feature	Recommendations/comments
Memory subsystem	• Powered by dual power circuit • SDRAM recommended • RDRAM devices target for the future designs • Memory devices are in self refresh state while in S3 • Voltage must be 3.0-3.6 V • Typical power per DIMM device 1-5 mA
PCI PME#	• Connected to pin A19 on all PCI connectors • Connected to pin dedicated for PCI wakeup event • All devices sharing the wake up pin must support PCI PM 1.0 (or later) specification

PCI Vaux	• Powered by dual power circuit (not just 3.3V stand by) • 375 mA expected by each PCI devices while in working state • sleep state wake up enabled – 375 mA / per device • sleep state wake up disable – 20 mA / per device • min/max current for system supporting one wakeup device is 435mA/1.5A
PCI reset	• Logic level high while in S0-S2 • Logic level low while in S3-S5 for proper support of D3 cold state

System reset and Power Good signal	• ACPI controller is reset only when AC power is applied
	• Memory controller is reset always after power to the controller has been removed
	  • Special case 1: power to the memory controller and memory array is kept while in S4 an S5 state. Power good to the controller is generated only when AC power is applied.
	  • Special case 2: power to the memory controller and memory array is removed while in S4 an S5 state. Power good to the controller is generated upon transition from S4 or S5 to S0. This signal can be generated based on the plane power. The RC and logic gate should be used to generate clean signal.

Signals to control power supply. SUSB# SUSC#	• Main power supply converter is turn off while in S3, S4 and S5 state • SUSB# signal from ACPI controller can be used to turn off the main converter. Ensure the proper polarity. • SUSC# signal is generated by the ACPI controller to indicate that system entered S4 or S5 state. It can be used to control power to the memory array.
Clock synthesizer and clock buffers	• All clocks in the system should be stopped until the main power converter is on and all voltages are within regulation. • "Power Good" signal from power supply can be used as qualifying signal
USB wakeup	• Requires 500 mA per port • When system is in S3 state the standby converter supplies power to the USB port • The power for USB needs to be from dual power source or standby converter has to have capacity needed for wakeup devices • Use 5 V dual to enable USB wakeup
Thermals	• Base designs have fan on is S0 and fan off in S3 • Use thermal sensor for active fan control • Use either linear voltage controlled fans or pulse width modulated fans to minimize noise while in S0 state • Use ACPI thermal policy for optimal performance
Minimize leakage currents due to split plane designs	• AGP Vref - voltage divider powered by the same the device itself. Dual voltage if part of the memory plane. Main voltage if part of the main 3.3 V plane. • GTL ref – connected to main power rail. Power not present while in S3 state. • VCC5_Vref – PIIX component (main 3.3V power is removed in S3 state) connect to 5 V main through 1 K resistor and to 3.3V main through the diode. Add decoupling capacitor to ground. • VCC5_Vref – BX component (3.3V power is maintained in S3 state) connect to 5 V main through 1 K resistor and to 3.3V dual through the diode. Add decoupling capacitor to ground. • If BX and PIIX components are on separate power planes the VCC5_Vref voltages need to be separated.

Table 6. 440BX design checklist

3. PCI Bus Power Managed Peripherals

ACPI provides the foundation for an Instantly Available PC. Desktop PC systems, characterized by add-in card expansion slots, now also benefit from the PCI Bus Power Management Interface Specification (PCI-PM). New add-in cards compliant with the specification dovetail seamlessly into an Instantly Available PC platform.

PCI functions compliant with the PCI-PM specification may also implement features that enable them to fully participate in the Instantly Available PC's "Off yet Communicating" capabilities. Key attributes include:

- Power Management Capabilities Reporting and Control
- Newly defined PCI connector pin for PCI Wakeup signaling
- Newly defined PCI connector pin for delivery of Auxiliary Power

PCI based network interface adapters, and modems that have been designed with a separate, split power plane for auxiliary power can fully support the Instantly Available PC's S3 (Suspend to Ram) state while also maintaining connectivity.

The PCI-PM specification is architecturally aligned with the ACPI specification, and is fully supported by ACPI-enabled operating systems developed by Microsoft, such as Windows 98 and Windows NT 5.0. For complete information on the PCI Bus Power Management Interface Specification, the reader is encouraged to obtain a copy of the specification from the PCI SIG web site. (http://www.pcisig.com)

3.1. Power Management Events

A power management event is any event, which triggers its PCI function to request a change of its power management state. A power management event is reported via the assertion of the PME# signal which has been assigned to pin A19 of the standard PCI Connector pinout. PME# is an output signal that is intended to be driven low by a PCI function to indicate that a power management event has occurred. Special design considerations apply to the design of an add-in card's PME# output driver in that the PME# network is shared between powered and unpowered PCI devices. Please refer to the PCI Bus Power Management Interface Specification for more information.

Typically a PCI function asserts PME# to request a change from a power savings state to its fully powered, and operational state. However, a function could assert PME# to request a change to a lower power state. The power management policies of the system, which reside in the operating system, ultimately dictate what action is taken as a result of a Power Management Event.

3.2. PCI "Off yet Communicating" Design Considerations

The PCI Bus Power Management Interface Specification, and the recent Auxiliary Power delivery ECR (currently before the PCI SIG for ratification) enable add-in functions to keep certain portions of their designs active even when the computer has been placed into a sleeping state such as S3. Entering a sleeping state, from the perspective of the PCI function is characterized by the powering down the PCI bus and all of its standard voltage rails. A new, standard auxiliary power source (3.3Vaux) is intended to be used by any PCI add-in card logic that would be required to wake the system if triggered to do so by some predefined stimulus. Modem designs, for example, could keep alive logic to sense the phone ringing and also possibly to decode and save caller ID information for later retrieval once the computer has been fully awakened.

With this new capability however, comes new design complexity associated with circuit designs that draw their power from both the main and auxiliary supplies. Regardless of whether a function asserts PME# from D3hot to request that it alone be resumed, or whether it requests that the entire system be awakened by asserting PME# from D3cold, a common set of issues must be comprehended at design time. Four classes of issues are discussed in this design guide:

1. Auxiliary Power
2. Preparing a PCI function for D3cold
3. Resuming from D3
4. Electrical issues

3.2.1. 3.3Vaux Auxiliary Power

The following subsections detail the 3.3Vaux requirements. The detailed description of 3.3Vaux can be found in PCI PM specification revision 1.1. The PME# and Vaux pins have also been added to PCI 2.2 specification.

3.2.1.1. Introduction to PCI 3.3Vaux

The PCI SIG Power Management Working Group has determined that in order to enable PCI bus power management to the full extent, PCI add in devices need a dedicated and guaranteed source of power to keep the wake event card logic active while the rest of the PCI bus is without power. The 3.3Vaux ECR defines a previously reserved connector pin (A14) as an auxiliary voltage supply to provide the standard source of power for PCI add-in card wake event logic.

3.3Vaux; 3.3V auxiliary voltage supply. This auxiliary voltage supply is optionally provided by the PCI system to pin A14 of the PCI expansion connectors. 3.3Vaux is used to power logic that needs to remain active when the rest of the system is unpowered (e.g. modem ring indicator detection logic). PCI functions that are enabled to draw current from this pin may consume no greater than 375mA. Functions that have not been enabled to draw fully from this pin must draw no greater than 20mA.

Several ***assumptions*** have been made in determining the optimal solution for supplying auxiliary power to PCI Add-In devices:

1. Add-In devices will need to function correctly in existing systems, which do not support 3.3Vaux, as well as new 3.3Vaux capable systems.
2. The majority of PCI add-in cards will not require 3.3Vaux (limited for the most part to communications peripheral subsystems e.g. NIC and Modem subsystems)
3. Due to item 2 above, systems should not be required to provide 3.3Vaux capacity to handle the worst case loading requirements

The first and third assumptions lead to the need for standard mechanisms to budget the total available 3.3Vaux power, and control the power consumption of 3.3Vaux capable add-in cards. Power consumption of 3.3Vaux capable add-in devices can be handled in one of 2 ways:

1. A motherboard could be designed to control 3.3Vaux current to individual slots based on software control
2. An add-in card could be made responsible for regulating it's own current consumption based on software control

The power management working group selected option 2 due to assumptions 1 and 2. Add-in boards will need to be able to control their usage of 3.3Vaux to operate in legacy systems as well as in 3.3Vaux capable systems,

and since the majority of cards will not need this capability, it is unreasonable to burden the entire system with the cost. Only cards which wish to consume 3.3Vaux power will carry the associated costs.

3.2.1.2. 3.3Vaux DC Characteristics

PC system manufacturers are required to physically route 3.3Vaux to Pin A14 of all PCI slots for a given PCI bus segment where 3.3Vaux is supported. Since most systems support a minimum of four PCI slots, it is essential to account for any auxiliary power required from the PCI slots and budget the available power appropriately.

The following table defines the DC operating environment that a 3.3Vaux enabled system must deliver.

Parameter	Min	Typ	Max	Units
3.3Vaux	3.0	3.3	3.6	Volts
$I_{MAX\_ENABLED}$	-	-	375 (note 1)	mA
$I_{MAX\_DISABLED}$	-	-	20 (note 2)	mA

Table 7. 3.3Vaux DC Characteristics

Note 1. Upper limit when function is in D0, D1, D2, D3hot or is in D3cold with its PME_En bit set.
Note 2. Upper limit when function is in D3cold and its PME_En bit is cleared to "0b"

Each 3.3Vaux enabled PCI add-in card's load on 3.3Vaux must not exceed 375 mA. This current requirement applies to the slot whether the load is applied for a single PCI function or multiple functions residing in a multi-function device. Like-wise, when a function has entered the D3cold state and its PME_En bit has not been set, (enabling it to generate PME#), it must then reduce its total slot 3.3Vaux current consumption to less than or equal to 20mA. Reducing total slot current consumption to less than or equal to 20mA can be done in a number of ways ranging from internally disabling as much logic as possible, to electrically isolating the 3.3Vaux pin from the auxiliary power plane of the add-in card.

In the case of a 4 PCI slot system for example, the minimum required 3.3Vaux current capacity that the system must be capable of delivering to the PCI slots (min requirement is for one PCI slot is enabled to wake the system at any given time) is 435 mA.

3.2.1.3. System Power Delivery Requirements

The optional 3.3Vaux pin must meet the following requirements if supported by the system:

1. 3.3Vaux must be connected to all motherboard PCI expansion slots.
2. The system must be capable of delivering up to 1.24W (375mA@3.3VDC) to each ***enabled*** PCI expansion slot.
3. The system must be capable of delivering up to 66mW (20mA@3.3VDC) to each ***disabled*** PCI expansion slot.
4. The system's auxiliary power source must be of sufficient capacity to support a minimum of one enabled PCI slot when the bus is in B3.

PCI functions in D0, D1, D2, or D3hot are unconditionally enabled to consume up to the 1.24W limit via 3.3Vaux. This is consistent with systems that are designed with dual mode power supplies. A dual mode power supply supports two separately tuned (load-wise) power sources for the same voltage reference. In the case of 3.3V, for example, the dual mode power supply will output both a high capacity, high efficiency 3.3V source for heavy "runtime" loads, and a lower capacity, yet reasonably efficient, 3.3V

source for lightly loaded "sleeping" states. 3.3Vaux, when supported by the system, must be connected to PCI connector pin A14, as the logical "OR" of the two supplies. In this way, under normally powered conditions (the PCI bus is in B0, B1 or B2) all PCI add-in cards, supporting $D3_{cold}$ functionality, connecting to pin A14 may draw up to 375mA from the 3.3Vaux pin. This voltage multiplexer could be implemented discretely on the motherboard with power switches (FETs) and associated control logic, or it could be integrated into the power supply itself. Regardless of the approach to implementation the voltage must never vary outside of the specified voltage regulation band (see Section 7.2. of the 3.3Vaux ECR) when switching between the main source and the lower capacity auxiliary source.

Software controls which slots are enabled for PME# generation from $D3_{cold}$ and is responsible for remaining within the system's auxiliary power budget

3.2.1.4. Add-in Card Power Consumption Requirements

When a system is switched from its main supply outputs (working state) to the auxiliary power source (sleep state), strict power budgeting with respect to which slots are allowed to consume full 3.3Vaux power becomes necessary. PCI functions must draw no more than 20mA through the 3.3Vaux pin when in D3cold unless enabled to do so as dictated by the state of their PME_En bit. The PME_En bit being low indicating that the function has not been enabled for PME# generation, in conjunction with the sticky" nature of the bit enable a straightforward implementation. Cards that have been designed to operate while in D3cold, yet have not been enabled to do so, must reduce the total current draw for the entire PCI slot to less than or equal to 20mA. This is accomplished by using the PME_En bit, if low, to gate off as much of the card's logic as is possible. This would include halting the internal clocking to all keep alive logic and any other means of reducing the card's total power consumption.

If a PCI function has been enabled for PME# generation prior to having entered into the D3cold state, the PCI add-in card (any single function, or combination of multiple functions) may continue to draw up to 375mA through the 3.3Vaux pin while in D3cold.

Auxiliary Power Consumption Reporting

The optional Data Register has been defined to enable the reporting of fine granular power utilization data for each of the supported power management states. In order to minimize the cost burden for designers while also ensuring a robust architecture, it has been decided to keep the Data Register optional, however a new (post PCI-PM revision 1.0) 3 bit field has been defined in the PMC register that is required for PCI functions that wish to draw from 3.3Vaux while in D3cold.

The bits, PMC(8:6), entitled "Aux_Current" provides a rudimentary vehicle for a PCI function to report its D3cold auxiliary power requirements to the system. The following table details the Aux_Current field.

08:06	Device Specific	Read Only	**Aux_Current -** This 3 bit field reports the **3.3Vaux** auxiliary current requirements for the PCI function. If the *Data Register* has been implemented by this function: • Reads of this field must return a value of "000b" • *Data Register* takes precedence over this field for **3.3Vaux** current requirement reporting. If **PME#** generation from ***D3cold*** is not supported by the function (*PMC(15)*=0), this field must return a value of "000b" when read. For functions that support **PME#** from ***D3cold***, and do not implement the *Data Register* the following bit assignments apply : **Bit 8 7 6 — 3.3Vaux Max. Current Required** 1 1 1 — 375 mA 1 1 0 — 320 mA 1 0 1 — 270 mA 1 0 0 — 220 mA 0 1 1 — 160 mA 0 1 0 — 100 mA 0 0 1 — 55 mA 0 0 0 — 0 (self powered)

Table 8 Aux _Current bit assignments

3.2.2. Preparing a PCI function for entering the D3cold state

Before the operating system places a PCI function in the D3cold state it must first ensure that all functional context is saved away to a non-volatile medium for later restoration, and that the function has been enabled for PME# generation (if the function is intended to be an "off yet communicating" resource). The following sequence illustrates the steps an ACPI-enabled operating system and device driver may follow in preparing a PCI function for the D3cold state. The following figures depict software elements in dashed boxes, and actual PCI hardware as the PCI bus and a solid line bounded PCI device.

Once the PM policy manager decides to place a function in the D3cold state, it dispatches its Class Driver to communicate this to the function's device driver.

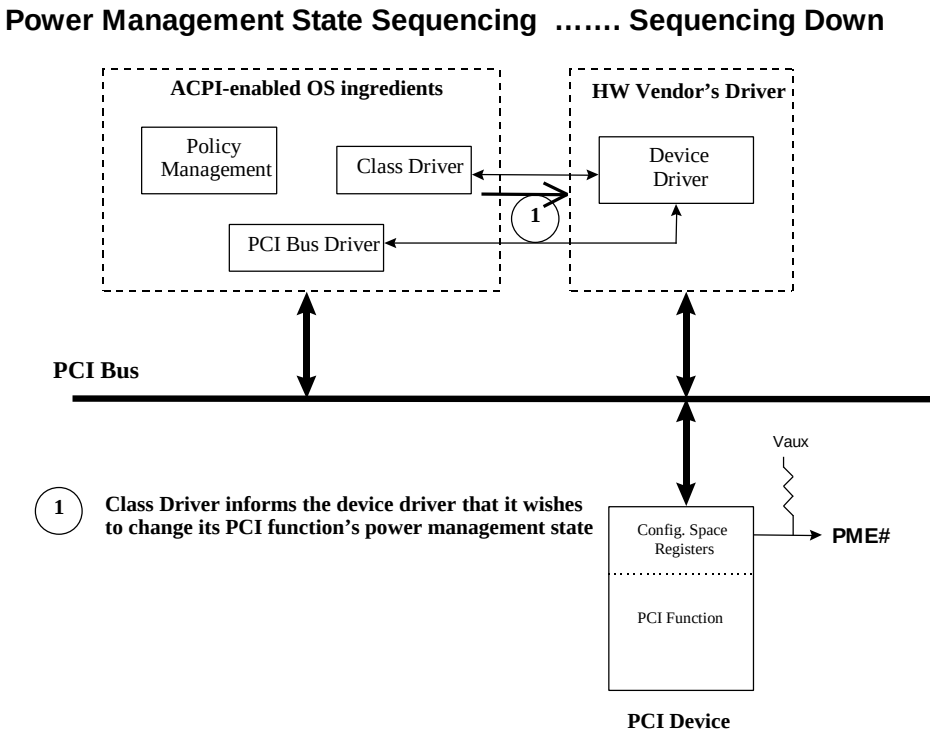

Figure 4. PM State Transitions…. Sequencing Down

The device driver then executes the appropriate code to save off any function context that would not otherwise survive the transition to the new PM state to a non-volatile storage medium, preferably auxiliary powered, self refreshing DRAM in support of the S3 Suspend to RAM state.

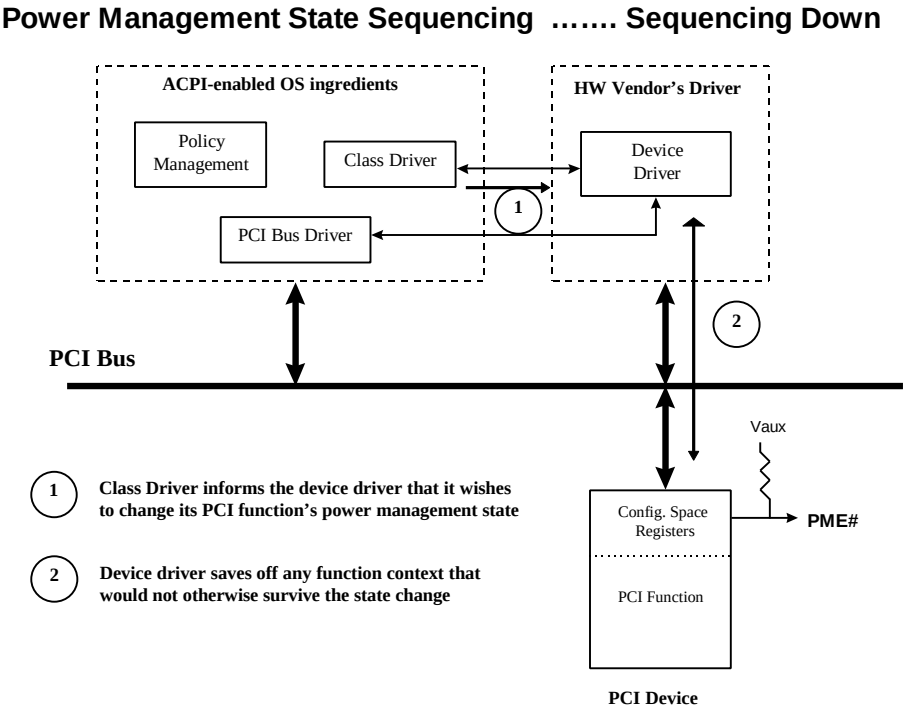

Figure 5. PM State Transitions…. (continued)

Once all context has been saved away the device driver returns control to the operating system which, by way of its PCI Bus driver, first enables the function to generate PME#, and then programs the D3 state into the function's PMCSR (Power Management Control and Status Register) PowerState field.

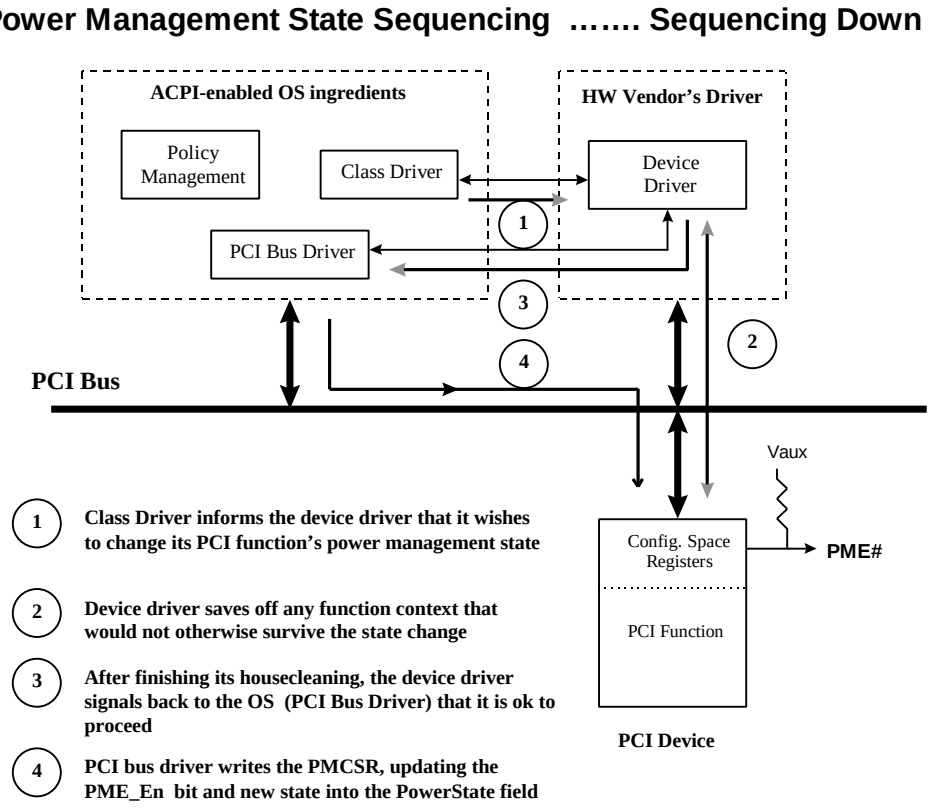

Figure 6. PM State Transitions…. (continued)

At this point the PCI function has been fully transitioned to the D3hot state, and it has been armed to wake the signal the host if triggered to do so by some predefined event.

In an Instantly Available PC with "off yet communicating" capabilities, the final step is to signal the DRAM subsystem to self-refresh itself, and to switch off the system's main power rails. The PCI function is now in the D3cold state and the system is in the "S3" state. The quick resume time (< 6 sec) characteristic of S3 (Suspend to RAM) enables the PC to remain connected even while the PC is "off" (sleeping).

3.2.3. Restoring a PCI function from the D3 state

Before the Operating System returns a function to D0 which will require a re-initialization of the function, it must ensure that it not only has the information necessary to re-initialize the function, but also any information necessary to restore it to its pre-sleep functional condition as well. This information is often client specific. As an example, assume a modem's client has set up a modem function in a specific state beyond its default initialization (error correction, baud rate, modulation characteristics, etc.). The client/function then goes unused for an extended period of time which may cause the PM policy manager to place the modem in the D3 state.

When the client is called upon to interact with the modem (e.g. in response to a ring-resume event), the Operating System will have transitioned the modem function to the D0 state. However restoration of the modem function to D0 alone may not be sufficient for the function and client to perform the indicated task. It is manifest

upon Power Management Device-Class specifications to include sufficient context save requirements for successful restoration of a function. The restoration must be transparent to the extent that the host application is unaware that a power state transition and the associated restoration had occurred.

3.2.3.1. Resuming from D3hot

Functions that are in the D3hot state have ether been placed there by the OS during the working state (having been determined to be an idle subsystem in an otherwise active system), or as a precursor to placing the system into a sleeping state (e.g. S3). There are two reasons for the function to be transitioned back to D0 from D3hot:

- User opens an application that requires the sleeping function to be in the D0, fully operation state
- An external event triggers the sleeping function to request that it be transitioned to the D0, fully operation state

The following sequence details the resume protocol from the D3hot sleeping state to the D0 fully operational state. The example sequence below illustrates the function having detected an external event which triggers its request to be transitioned to D0. Note that essentially the same sequence of events also occurs when the operating system wishes to reactivate the function, the only difference being that the function would not have asserted PME# to initiate the sequence in that instance.

Following the detection of an external event of interest the PCI function asserts PME# to the system requesting that the operating system transition it back to D0.

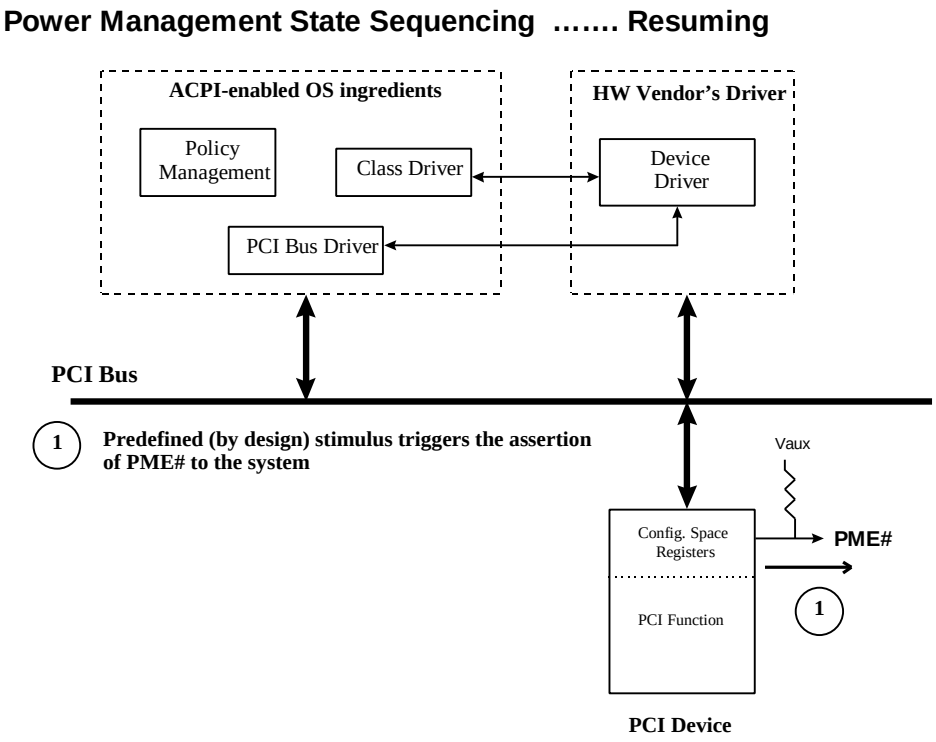

Figure 7. PM State Transitions…..Resuming to D0

At this point the operating system dispatches the PCI bus driver to determine which PME enabled function(s) had generated the current Power Management Event. This is done by reading each of the PME enabled PCI functions' PMC register, specifically the PME_Status bit of each. PME_Status being set to "1" indicates that the function has experienced a Power Management Event. Assuming that the operating system has determined that the subject function had been the source of the current PME#, the PCI bus driver would then program the function back to the D0 uninitialized state. Note that the function is uninitialized at this point because all of its context, with the exception of its PME context, had previously been saved off to a non-volatile medium, and is assumed to have been lost internally in order to enable the function to achieve the maximum power savings while in D3hot.

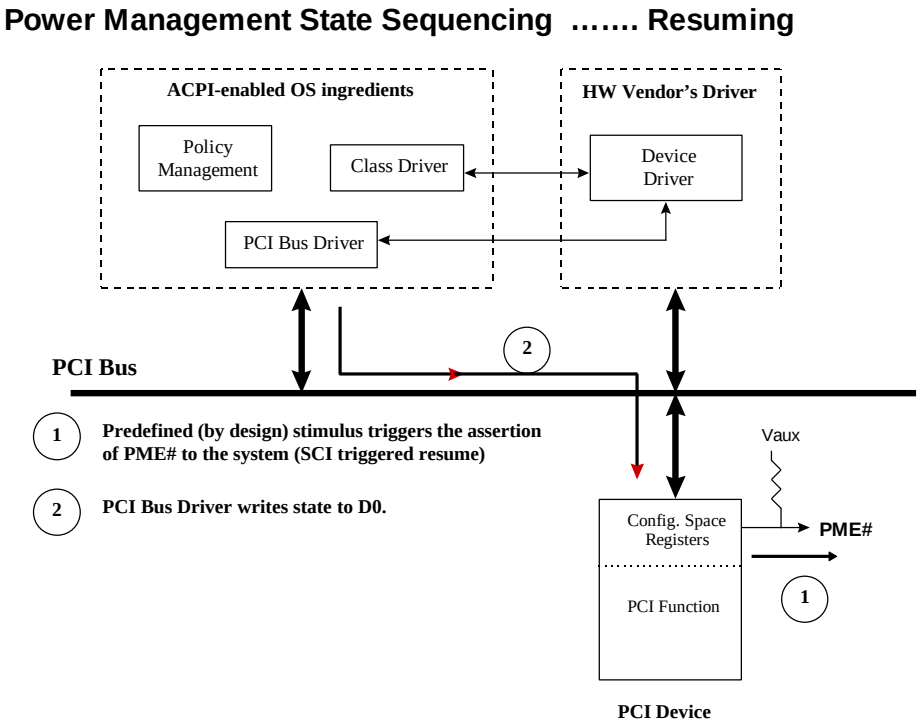

Figure 8. PM State Transitions….(continued)

Following this, control is passed to the class driver which commands the device driver to restore all functional context that had previously been stored away. Once the device driver has restored the function's context, it then passes control back to the class driver signifying that the function has been restored to the D0active, fully operational state.

Were this function a modem, the phone ringing would have precipitated the previous sequence of events followed almost certainly once in the D0active state, by the modem's assertion of a functional interrupt (PIRQ) request for service of its communications port.

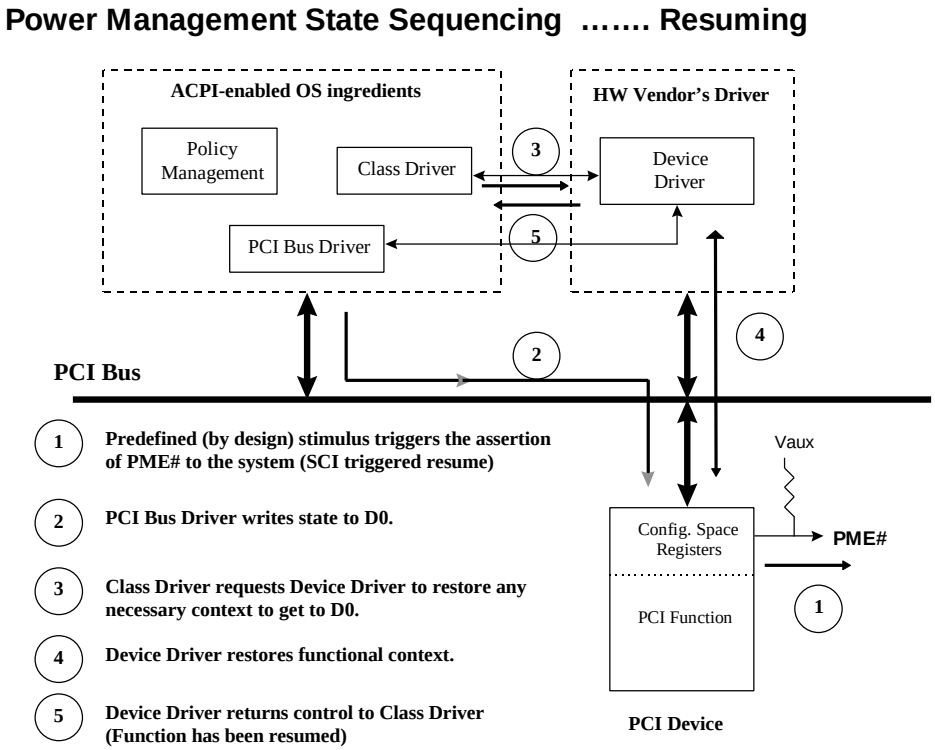

Figure 9. PM State Transitions…. (continued)

3.2.3.2. Resuming from D3cold

When a function is in the D3cold state its host system is typically in its lowest power sleeping state (e.g. S3). The bus segment that the PCI function resides on is always in the B3 state. When a function is in the D3cold state the only means of transitioning it back to the D0active, fully operational state is by reapplying power to the bus and asserting PCI RST#. This is essentially the power on PCI bus reset sequence. There are some differences, however, which arise from the fact that some portion of the PCI function had remained powered via auxiliary power (3.3Vaux).

A fundamental concern, associated with this aspect of a PCI-PM compliant design, is in determining how to observe, reliably, the assertion of PCI RST# from the perspective of the 3.3Vaux powered logic. Since the PCI bus is in B3, the 3.3Vaux powered logic must be able to tell the difference between the legitimate assertion of PCI RST# (following reapplication of PCI bus power), and an unpowered bus with RST# floating at or near ground potential. The card designer could go to the added expense of developing his own onboard Vcc "power good" detection scheme by monitoring the main PCI bus voltage rails, however a simpler and less expensive

alternative is being pursued. A new, requirement affecting the behavior of the RST# signal will be imposed upon new PCI-PM compliant systems that support the delivery of 3.3Vaux to their PCI slots.

New PCI RST# Requirements

The section entitled "Reset" of the PCI Local Bus Specification requires the assertion of RST# whenever the PCI main power rails are out of spec, either when low and ramping on, or low as a result of a power failure event. In systems that do not support 3.3Vaux, when a power failure event is occurring the assertion of RST# causes all PCI devices' output buffers to be floated until such time that power is completely lost. At this point the entire bus is floating, including the RST# signal.

With the addition of "programmable power failures", and the 3.3Vaux power source, two new requirements for RST# are added to the existing requirements..

- ***The central resource of systems that support 3.3Vaux must ensure that the PCI RST# signal remains asserted whenever the PCI bus is in the B3 state.***

This can be accomplished either by powering the PCI RST# output buffer with auxiliary power, or by deploying a weak pulldown resistor on the RST# signal (on the motherboard) such that when power is lost at the RST# output buffer the signal network will have a low impedance path to ground. With RST# asserted whenever the bus is in B3, PCI functions that are in the D3cold state can depend on the low to high transition of RST# as a reliable indication that they must internally initialize all volatile portions of their function and transition themselves to the D0uninitialized state.

The following diagram illustrates the required behavior of RST# when the bus transitions into, and out of B3. Refer to the "Reset" section of the PCI Local Bus Specification for existing RST# requirements.

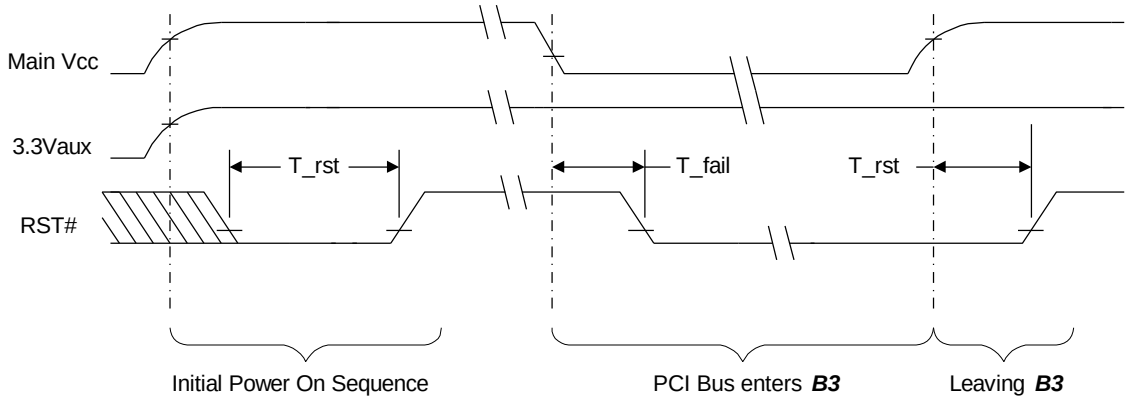

Figure 10. *B3* Reset Timing

- ***Functions in $D3_{cold}$ that are capable of generating PME# from $D3_{cold}$ must also be capable of accepting PCI bus cycles, including PCI configuration cycles, no later than 10mS following the low to high transition of the PCI RST# signal***

PCI functions that wake the system with power management events are designed specifically for quick resumption, and must take no longer than 10mS to self initialize to the point where they are ready to accept PCI cycles targeting them.

From the D0uninitialized state the OS (class driver) commands the device driver to restore all function context required for the PCI function to assume the same functional status that it had exhibited just prior to the point in time when it was programmed to the D3hot state. Once all function context has been restored to the function, thereby transitioning it to the D0active state, control is passed back to the OS.

3.2.3.3. Isolation of 3.3Vaux from Main 3.3V

When a PCI add-in card is plugged into a system that supports 3.3Vaux, the card must ensure that 3.3Vaux is electrically isolated from the main PCI 3.3V rails at all times. This is done via split power plane PCB design with a separate power plane for 3.3Vaux which never comes into electrical contact with the add-in card's main PCI 3.3V power plane.

In designs where logic paths cross between components powered by the 3.3Vaux and 3.3V power domains, special care must be taken to ensure that when power is removed from the main 3.3V rail that no physical damage, or logic malfunction occurs with respect to any of the subject devices be they powered or unpowered. The add-in card must electrically isolate these cross-domain powered circuit paths from each other as a direct result of being programmed to the D3hot state. At this point all PCI power rails are still within their specified voltage regulation band, and the D3hot state is always the initial step towards the eventual removal of main PCI bus power rails. (i.e. to B3/D3cold)

3.2.3.4. Voltage Sequencing

The 3.3Vaux power source is independent of the PCI bus's 3.3V and 5V voltage rails, and there are no voltage timing sequencing requirements between them. This is consistent with the sequencing requirements in the PCI Revision 2.1 specification. (refer to Section 4.3.4.2)

Upon powering up the system all three power rails (if supported) ramp to their specified voltage levels independent of one another with respect to timing. Once all supported voltages have settled to their specified levels 3.3Vaux will remain within its specified levels at all times while main PCI voltage rails 3.3Vcc and 5Vcc may be switched on and off periodically under program control.

3.3. PCI-based "Off yet Communicating" Subsystems

3.3.1. Power Managed PCI LAN Controller

The power management of LAN devices is described in the *Network Device Class Power Management Reference Specification.* (http://www.microsoft.com/hwdev/download/netpmspc.rtf)

Network Wake-up Events: There are three main categories of network device wake-up events

Programmable Pattern filtering wake-up event: The ideal wake-up on LAN scenario is one where a PC that's in a power reduced "sleep" state ***transparently*** wakes up when something else on the LAN or in the enterprise tries to connect to it. Except for slight delays during the waking process, the other machines attempting to access or manage a sleeping machine's content would be unaware that the target PC is being power managed.

Unfortunately, today's networking protocols generate a significant amount of network traffic (broadcast queries, etc.) that machines connected to the LAN must monitor and then respond to if necessary. The majority of this traffic is targeted for the other (N-1) machines connected to the LAN. To make matters worse, it often requires work by upper protocol layer software running on these machines' processors to determine whether or not one of

these packets are of any interest. This does not align well with power management, where it is desirable to put those same PCs (and processors) into a sleep state.

To permit the PC and its processor to enter a power-managed sleep state, new capabilities have therefore been defined where the LAN device is programmed by upper software layers with "interesting" packets that should wake-up the PC for the purposes of making a desired connection. When put into this filtering mode, all other "non-interesting" packets are ignored.

The number and size of these packet pattern filters can vary among various network device designs, and the OS negotiates and adapts as necessary. Certain designs may be cost-optimized for a particular networking environment (e.g., Corporate LAN with a WINS server), and may only support a few patterns of limited depth. Other designs may be more versatile, providing more and deeper patterns that make it flexible for operating in more complex networking environments, as well as extensible for use with future protocols (multiple simultaneous VLANs, IPV6, etc.) as they become significant. The pattern matching capabilities of a particular solution should be evaluated carefully to assure they are suitable for the targeted networking environment.

Another significant ***option*** for network devices is whether or not the packet that caused wake-up of the PC is retained or not. If the wake-up packet is NOT retained by the network device, network protocols (and perhaps their overlying applications) may have to retry the PC, which will respond once awake. Network devices that CAN retain the wake-up packet may significantly reduce the access time and improve the reliability of incoming connections. Again, these cost vs. performance tradeoffs should be studied carefully.

The device states (D0, D1, D2, D3) from which a network device can assert pattern-filtering wake-up may vary from one design to another. Designs that can cause wake-up from the lower power device states (e.g., D3cold) are less dependent on system/bus resources, and enable the entire PC to sleep in a lower power state as a result. For example, devices that can leverage an auxiliary power source and don't need PCI bus clock to perform wake-up will enable a system wake-up on a LAN event while in the ACPI S3 state (Suspend To RAM).

Magic Packet™ wake-up event: This simple wake-up on LAN scheme uses a special data pattern unique to that LAN device that can be easily identified at minimal cost (multiple copies of its unique HW MAC address) . It was intended to allow management applications to remotely power up (cold boot) a PC to perform management functions (virus scans, inventory, driver/OS updates, etc.). The benefits of this scheme are that it is simple to implement, and is often able to function independent of the OS. It falls short however in that it does not enable transparent access – only management apps specially designed to discover and generate these unique data patterns work.

Magic Packet™ has been "grandfathered" into the network device class specification as an ***option*** so that existing applications that use it are able to wake-up the PC for management from ACPI sleep states, in addition to when the PC is "OFF".

Magic Packet™ also adds significant value beyond OSPM solutions, in that it is possible to wake-up and manage a PC from a "legacy" soft-OFF state when the OSPM capable operating system has yet been loaded (or installed). System designers should carefully evaluate what special power supply routing and control mechanisms are required to additionally deliver Magic Packet™ power up from soft-OFF states.

Other utility wake-up events

- Detection of a change in the network link state

  The goal of link state change being a wake-up event is so that a sleeping system that is waiting for a network wake-up packet can be notified when the link is down (i.e. no wake-up packet can be sent to the system) so that it can lower the device's power state and, perhaps, the system's sleeping state in order to conserve energy. Note that the Link Down condition

may be the result of temporary wiring changes on the net, and not necessarily the disconnection of a cable from the network device itself. Because of this, an appropriate time-out should be used before deciding to lower the device's power state.

- Vendor defined wake-up events

  Manufacturers may also implement other types of wake-up events that are not listed here, using private interfaces. Network devices that support wake-up events must have device-specific methods for individually enabling and disabling each wake-up source, and for indicating which source caused a wake-up

3.3.2. Power Managed PCI Modem

The power management of communications devices (UARTs and modems) is described in the Communications Device Class Power Management Reference Specification. This specification is available for download at: (http://www.microsoft.com/hwdev/download/compmspc.rtf) There are three key functional aspects to the power management of modems:

1. Modem device power conservation based on the activity of communications applications
2. PC wake-up on ring for incoming calls
3. PC wake-up on ring with caller-ID preserved.

Simple modem device power conservation: When the PC is in its working state and the OS detects that a communications application has interest in either placing outbound calls or receiving inbound calls, the modem will be left "***on"*** in its fully-on D0 state. Otherwise the modem will be shut ***"off"*** to its D3*hot* state.

Wake-up on ring: As the PC transitions to a sleep state, the modem will be enabled to wake-up the PC on ring if the OS detects that a communications application may be interested in receiving incoming calls. The modem will be put into the lowest power device state (e.g. D2 or D3) from which it has reported support for wake-up capabilities.

The D2 state is defined such that a modem can conserve power by turning off all circuitry that is not essential to detecting ring and asserting a wakeup signal. Most designs should be able to shut down power or local clocks to the modem's digital signal processing or embedded controller functions while in this state, thereby requiring minimal power to detect the ring.

Although the D3 state is primarily used to shut the modem "off " when not in use, there is another extremely important usage for D3. A modem can also be designed to support PC wake-up from D3***cold***, where the bus (e.g., PCI) connecting the modem to the PC is not providing primary power. Modems that have this capability permit the PC to sleep in much lower power states (e.g. S3). Because the bus is not powered, the modem must be designed to obtain the power required to detect ring and cause wake-up from another auxiliary source. For PCI add-in cards, the preferred source of auxiliary power is a newly defined 3.3Vaux pin (pin A14) for the PCI bus connector.

Wake-up on ring with caller-ID preserved: Caller-ID signaling is transient, and appears only when the phone is just starting to ring. Because caller-ID reporting would be missed by PCs while still in the process of waking from a sleeping state, the ability for a modem to retain caller-ID while in D2 or D3, and then repeat it later on demand when the PC has awakened is ***strongly recommended***. The mechanism for doing this is described in the *Communications Device Class Power Management Reference Specification*, with the supporting command syntax described in the V.voice and TIA-695 voice modem specifications.

The following figure illustrates the real-time nature of the Caller ID message. Note that in Europe the Caller ID message comes in PRIOR to the first ring.

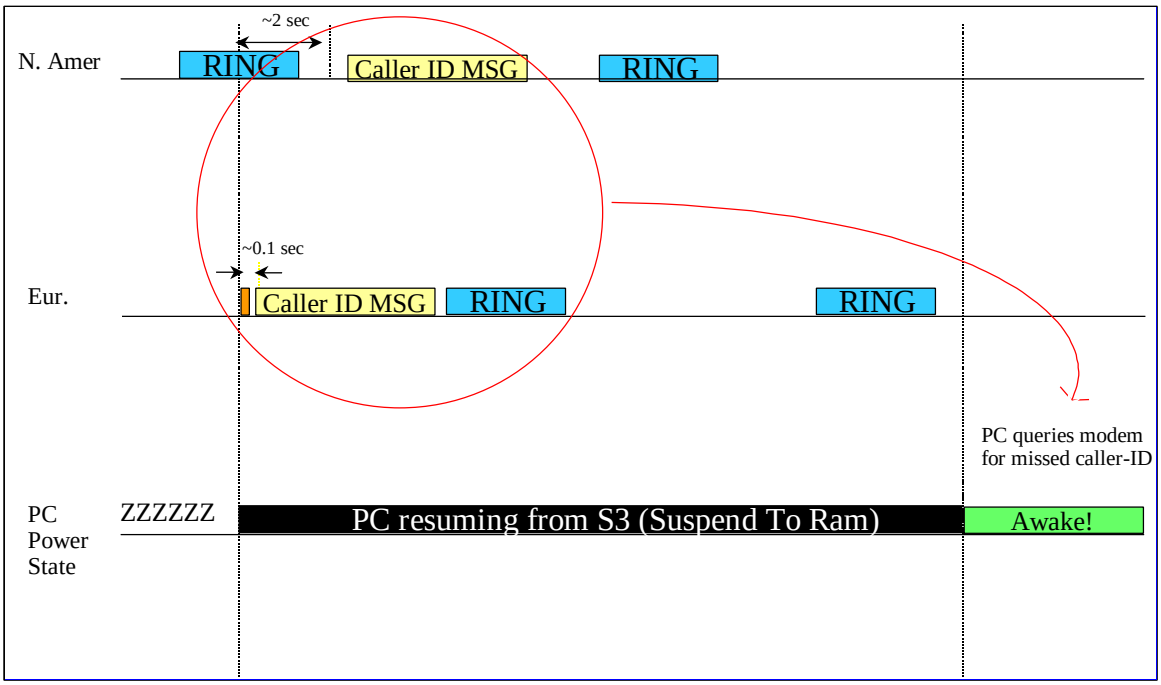

Figure 11. Caller ID Timing

What becomes clear from Figure 10 is that in order to maintain "Off yet Communicating" capabilities the quick (~4-6 seconds) resume time associated with the S3 Suspend to Ram state is a requirement. In most cases a sleeping system in S3 will be fully resumed by the end of the second ring. By contrast, a system implementing S4 (Suspend to Disk) could take as long as a minute to fully resume. Even with Caller ID decoding and capture in the modem hardware, S4's long resume time will almost certainly lose even the most persistent caller who will likely hang up after 8 or fewer rings.

Designing modems to support wake up from the D3cold state is strongly recommended. Breakthroughs in the level of power savings and user experience come only with the ability to shut down most of the PCs power, fans and other sources of noise. Unless the modem can wake the system from D3cold it will limit the rest of the system from achieving these objectives.

3.4. Power Managed AC'97 Audio

Any audio subsystem that targets motherboard integrated implementations exclusively should ensure that their design can be implemented with an ACPI compliant motherboard/BIOS. However it is reasonable to assume that in most, if not all cases the audio vendor would not wish to limit his products' marketability to only motherboard design wins. As such it is highly recommended that all new PCI audio designs implement the PCI Power Management interfaces.

Once an audio vendor has committed their product designs to the PCI power management interfaces these same products may be implemented as either motherboard or add-in solutions. And, in the case of motherboard integrated solutions, these PCI-PM compliant products would not impose a software burden on their OEM customers. ACPI compliant motherboard devices must be comprehended by the motherboard BIOS. PCI-PM compliant designs, on the other hand, rely solely on the device driver that ships with the audio hardware, as well

as the power management aware PCI bus driver that ships with an ACPI enabled operating system such as Window 98 or Windows NT 5.0.

3.4.1. Recommended PCI Power Management Support for Audio

The PCI Bus Power Management Interface Specification requires that a compliant PCI function, at a minimum supports the D0, D3hot, and D3cold states. These minimum requirements are adequate for audio only subsystem implementations. Additionally, it is not deemed necessary for an audio subsystem to support PME# wake capabilities.

New AC '97 architectural extensions ease the integration of the modem analog front end with audio in the same physical package. In doing so, however, power management issues arise which must be fully comprehended at design time.

The Instantly Available PC implements a dual mode power supply (see Section 5) that is intended to support specific "always active" functions while the majority of the PC is powered down. 5.0V *and* 3.3V auxiliary power sources are available on the motherboard, and a 3.3Vaux supply pin is currently being added to the PCI connector definition.

When a modem AFE codec is integrated along with an audio codec into a single device, and the usage expectation is that the modem be used as an "off yet communicating" resource, the power delivered to the codec must be from the auxiliary voltage supply. This could impact the upper limit of audio performance as described in the following section.

3.4.1.1. Combined Audio/Modem AFE Codec (AMC '97)

For AMC '97 combined Audio/Modem AFE implementations the codec, AC-link and portions of the digital controller all must be powered by Vaux as illustrated below:

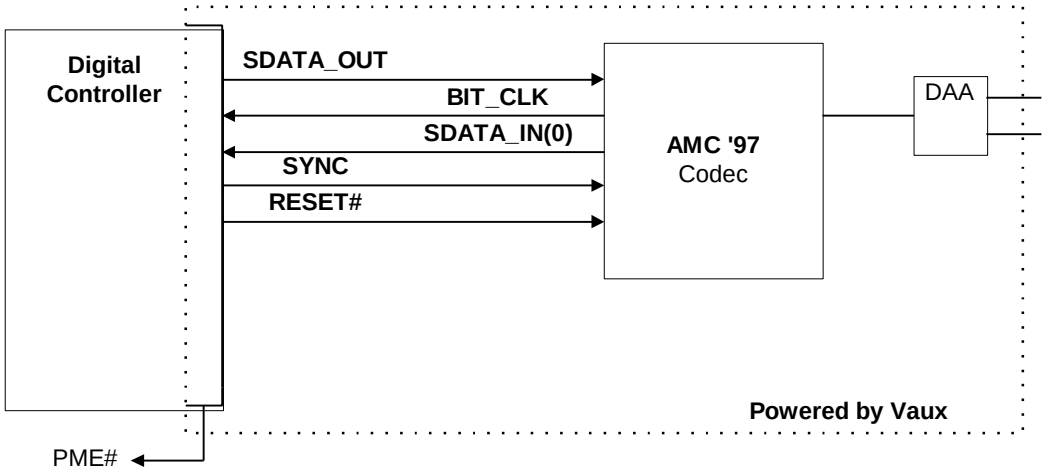

Figure 12. Combined Audio / Modem AFE Auxiliary Power Distribution

The codec and AC-link are programmed to a low power state (see Figure 12) and, upon detection of a power management event, are brought back to the active state by executing a warm reset sequence as detailed in Figure 14.

Figure 12 below shows the AC '97 digital controller placing the AC-link into its lowest power state by programming the codec's Powerdown Control/Status register with bit(12) = 1 (PR4) .

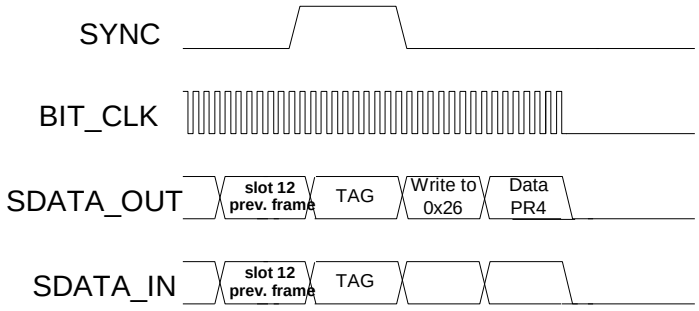

Figure 13. AC-link Low Power Mode

In response to this command BIT_CLK and SDATA_IN codec, and SDATA_OUT controller outputs go low and stay low .

AC-link when programmed to its low power mode, can only be reactivated by the device driver which can write to an AC '97 digital controller register causing it to signal a cold or warm reset on the AC-link. A warm reset, which will not alter the current AC '97 registers, is signaled by driving SYNC high for a minimum of 1uS in the absence of BIT_CLK.

Within normal audio frames SYNC is a synchronous codec input. However, in the absence of BIT_CLK, SYNC is treated as an asynchronous input used to signal a warm reset to the AC '97 codec.

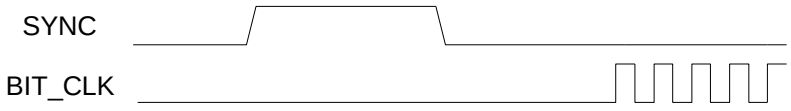

Figure 14. AC-link Warm Reset

In an AMC'97 implementation, where the audio/modem AFE codec and AC-link are both completely powered by Vaux, an enabled power management event detected at the modem interface causes the assertion of the PME# signal to the system. PME# assertion causes the system to resume so that the modem event can be serviced. The first thing that the device driver must do to reestablish communications with the codec is to command the DC'98 to execute a warm reset to the AC-link. The entire sequence is illustrated in Figure 14 below.

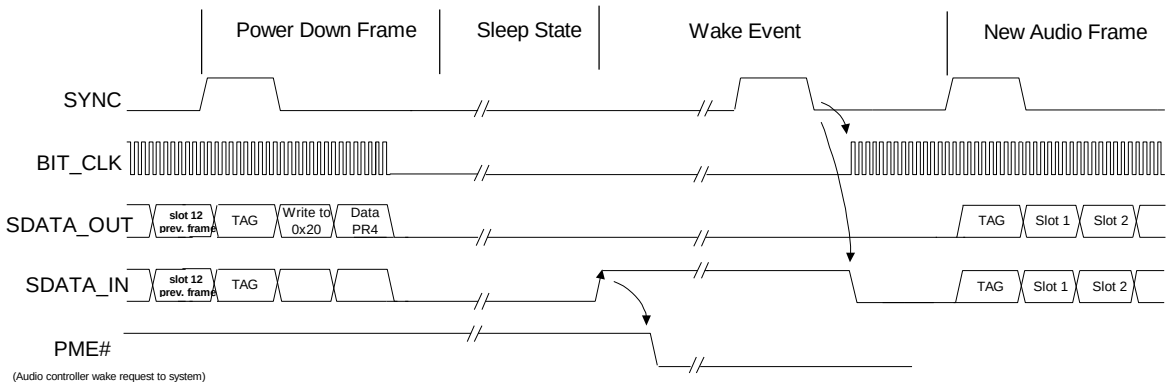

Figure 15. AC-link Power Down/Up Sequence

The rising edge of SDATA_IN causes the AC '97 digital controller to assert its PME# to the system's ACPI controller. The AMC '97 codec must keep SDATA_IN high until it has sampled SYNC having gone high, and then low. PME# is cleared out in the AC '97 digital controller by system software, asynchronous to AC-link activity. The AC '97 digital controller must always monitor the codec's ready bit before sending data to it.

While the combined audio/modem AFE codec brings the lowest cost, and smallest mechanical footprint (ideal for mobile computers), it does require that the audio codec also be powered from the auxiliary voltage source which may not be as tightly regulated as the audio codec might like. The system designer should carefully evaluate the quality of the dual mode power supply he chooses if his target design requires lowest combined cost for both audio and modem, "off yet communicating" capabilities, AND highest audio performance.

While the combined codec solution is ideal for footprint constrained mobile and very low cost desktop PC designs, a more scaleable split partitioned design could serve the midrange to high end market segments well. The next section discusses a split partitioned AC-link, another AC'97 architectural extension rolled out as part of Intel's Audio'98 roadmap.

3.4.1.2. Split Partitioned Implementations (AC '97 2.0 + MC '97)

The split partitioning allows for independent, tightly regulated power to be applied to the audio codec, while also enabling the modem interface to support "off yet communicating" capabilities.

In a split partitioned implementation, where separate audio and modem AFE codecs are employed, the MC '97 codec, its DAA, a common clock oscillator, and portions of the AC '97 digital controller are powered by Vaux. The AC '97 audio-only codec is powered via its normal DVdd, and AVdd supplies, and as such is shut completely off when the system enters a sleeping state. The following figure illustrates an example of a split partitioned codec implementation.

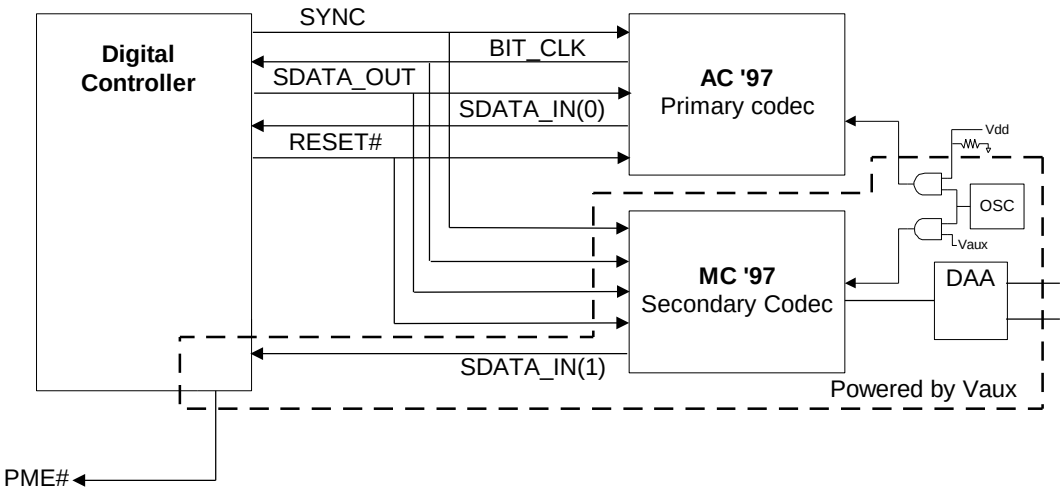

Figure 16. Example Split Partitioned Audio/Modem Design

Once the system enters a sleeping state Vdd shuts off which causes the oscillator input to the AC'97 to go low and remain low. Clocking remains active at the MC '97 which continues to look for ring detection and Caller ID data while the rest of the system sleeps. A wake, or power management event causes the MC'97 to transition its SDATA_IN from low to high which in turns causes the AC '97 digital controller to assert PME# to the system. Once the main power has been reapplied the device driver executes a cold, or warm, AC-link reset followed by the restoration of any saved off functional context. Following SDATA_IN's low to high transition as a result of a Power Management event, it must remain high until either a warm or cold reset is observed on the AC-Link. After sampling either a cold or warm reset, the modem codec must drive SDATA_IN low and wait for the first new audio/modem frame to begin.

The MLNK bit

Miscellaneous Modem Status and Control Register (56h) Bit 12.

Bit 12 of the Misc. Modem Status and Control Register (MLINK), when set to a "1" indicates that the AC '97 digital controller is about to program the AC-link to the PR4 sleep state (as defined in section 5.2 of AC '97 rev 1.03).

Once this bit has been set, the MC '97 codec must drive and hold its SDATA_IN line at a logic low "0" level until one of three things happen:

1. A RESET# is asserted on the AC-link

2. Warm Reset sequence signaled on the AC-link
3. A power management event occurs, such as a ring detection

The sampling of RESET# (Cold) assertion on the AC-link must trigger the same codec behavior as is currently specified in Section 5.2.1.1. of AC '97 rev 1.03.

If the modem codec observes a Warm Reset sequence (i.e. SYNC assertion in the absence of BIT_CLK) then the modem codec shall awaken in the manner specified in Section 5.2.1.2. of AC '97 rev 1.03.

If the modem codec has been enabled to wake the system, and a power management event occurs (such as the phone ringing), then the modem codec shall comply with the behavior specified in Section B.5.2 of the AC '97 rev 2.0 extensions specification.

For further details the reader is referred to Intel's Audio '98 web page located at:

http://developer.intel.com/pc-supp/platform/aud98/

3.5. Power Managed Graphics Controller

It is highly recommended that the system designer build Instantly Available PCs using AGP-based graphics subsystems.

AGP graphics delivers significant benefits at the platform level including:

1. Power Manageable via standard PCI-PM interfaces
2. Highest performance
3. Frees up significant headroom on the PCI bus for other demanding I/O bandwidth consumers

AGP controllers are enumerated as PCI agents with enhanced capabilities. They are the first PCI function to have potentially two entries in their Capabilities List. AGP controllers provide a Capabilities List entry to inform the OS of the presence of AGP functionality, and may also have an entry to support the PCI Bus Power Management Interface Specification.

System designers and integrators are urged to deploy only those AGP controllers that are also compliant with the PCI Bus Power Management Interface Specification.

In addition to the minimum set of PCI-PM requirements, either D1 or D2 should also be supported by the graphics controller. Support for one of these intermediate states enables a power reduced mode with quick resume time which could be used while the PC is in the working state (e.g. a quick resume "screensaver" mode).

PME# wakeup support by the graphics subsystem is not deemed necessary. While supporting the actual generation of PME# may not be necessary, it could be advantageous for an AGP controller to report that it does support PME# from D3cold for other reasons. When a function is resumed from the D3cold state it must restore its full context prior to its resuming normal operation. In the case of a graphics controller, restoring context would also mean restoring its local frame buffer memory which if large enough could have a significant impact on the resume latency (visual) of the system.

If an AGP graphics subsystem supported self-refresh with a split plane PCB design it could, if it reported support for PME# from D3cold, connect the split DRAM plane to the 3.3Vaux pin of the AGP connector. In doing so it could, even if not enabled for PME#, consume up to 20 mA when the system is in S3. This would be enough current to keep roughly 32MB of local SDRAM frame buffer memory refreshed. The system designer needs to

determine the minimum frame buffer memory footprint size where the resume latency impact would be great enough to justify keeping the private frame buffer self-refreshed.

4. ACPI BIOS Design Considerations

The primary role of the ACPI BIOS is to supply the ACPI tables, which describe a particular platform's hardware to the operating system (OS). Other responsibilities include modifying the system memory map function call to account for the memory occupied by the ACPI tables, post modifications to restore context when waking, and Power Management mode control to switch between ACPI and Legacy mode.

Additions/changes that required to add ACPI support to legacy BIOS can be summarized in the following categories:

- ACPI Tables
- Memory reporting
- POST Modifications (Resume flow)
- Power Management mode control (Enable/Disable ACPI)
- Save to Disk
- Setup
- PC98 requirements (Fast BIOS Post)

4.1. ACPI Tables

ACPI uses tables to describe system information, features, and methods for controlling those features. These tables list devices on the system board or devices that cannot be detected or power managed using some other hardware standard. They also list system capabilities such as the sleeping power states supported, a description of the power planes and clock sources available in the system, batteries, system indicator lights, and so on. This enables the ACPI driver to control system devices without needing to know how the system controls are implemented.

4.1.1. Root System Description Pointer ("RSD PTR")

This is the first structure the operating system examines when building information from the ACPI tables. Located in low memory, the only purpose of the Root System Description Pointer is to provide the address of the Root System Description Table.

- Points to the Root System Description Table ("RSDT")
- Located either in the first 1K of the Extended BIOS Data Area, or in the memory space between 0E0000h and 0FFFFFh

4.1.2. Root System Description Table ("RSDT")

Like all System Description Table structures, the Root System Description Table begins with the table's signature, length, revision and checksum, followed by details about the OEM, and utility that created the table. Following the table header, the Root System Description Table provides an array of pointers to other System Description Tables.

- Points to the Fixed ACPI Description table ("FACP")
- Points to the Secondary System Descriptions Table ("SSDT")

- Points to the Persistent System Description Table (“PSDT”)
- Points to the Multiple APIC Description Table (“APIC”)
- Points to the Smart Battery Table (“SBST”)
- Located in ACPI Reclaim Memory

4.1.3. Fixed ACPI Description Table (“FACP”)

In addition to providing pointers to other System Description Tables, the Fixed ACPI Description Table defines various fixed ACPI information including the system port addresses of the hardware register blocks, details on flushing the processor’s caches, and real time clock alarm functionality.

- Points to the Firmware ACPI Control Structure (“FACS”)
- Points to the Differentiated System Description Table (“DSDT”)
- Describes the implementation and configuration details of the ACPI hardware registers
- Located in ACPI Reclaim Memory

4.1.4. Firmware ACPI Control Structure (“FACS”)

The Firmware ACPI Control Structure is located in ACPI NVS memory, a memory range set aside by the BIOS where the OS and Firmware can exchange control information. Included in this structure is the hardware signature, the firmware waking vector, the global lock, and the firmware control structure flags.

4.1.5. Differentiated System Description Table (“DSDT”)

This table, located in ACPI Reclaim memory, contains implementation and configuration information the operating system can use to perform Power Management, thermal management, or Plug and Play functionality that goes beyond the information described by the ACPI hardware registers. These implementation details are provided in the form of objects that contain data, AML code, or other objects.

The following tables are not required. Their definition’s can be found in section 5.2.7 of the ACPI specification:

- Secondary System Descriptions Table (“SSDT”)
- Persistent System Description Table (“PSDT”)
- Multiple APIC Description Table (“APIC”)
- Smart Battery Table (“SBST”)

4.2. Memory Reporting

With the addition of the ACPI tables, the IA-PC INT15 E820 specification has been updated with two new memory range types, ACPI Reclaim Memory and ACPI Non-volatile Sleep Memory.

4.2.1. ACPI Reclaim Memory

Memory identified by the BIOS that contains the ACPI tables. This memory can be any memory address region above 1 MB. When the OS is finished using the ACPI tables, it is free to reclaim this memory for system software use (application space).

4.2.2. ACPI Non-Volatile-Sleeping Memory (NVS)

Memory identified by the BIOS as being reserved by the BIOS for its use. Except as directed by control methods, the OS is not allowed to use this physical memory.

To illustrate the new memory ranges, the BIOS will report the system memory map by E820 as shown in Figure 16. [Note that the memory range from 1 MB to top of memory is marked as system memory, and then a small range is additionally marked as ACPI reclaim memory.] A legacy OS that does not support the E820 extensions will ignore the extended memory range calls and correctly mark that memory as system memory.

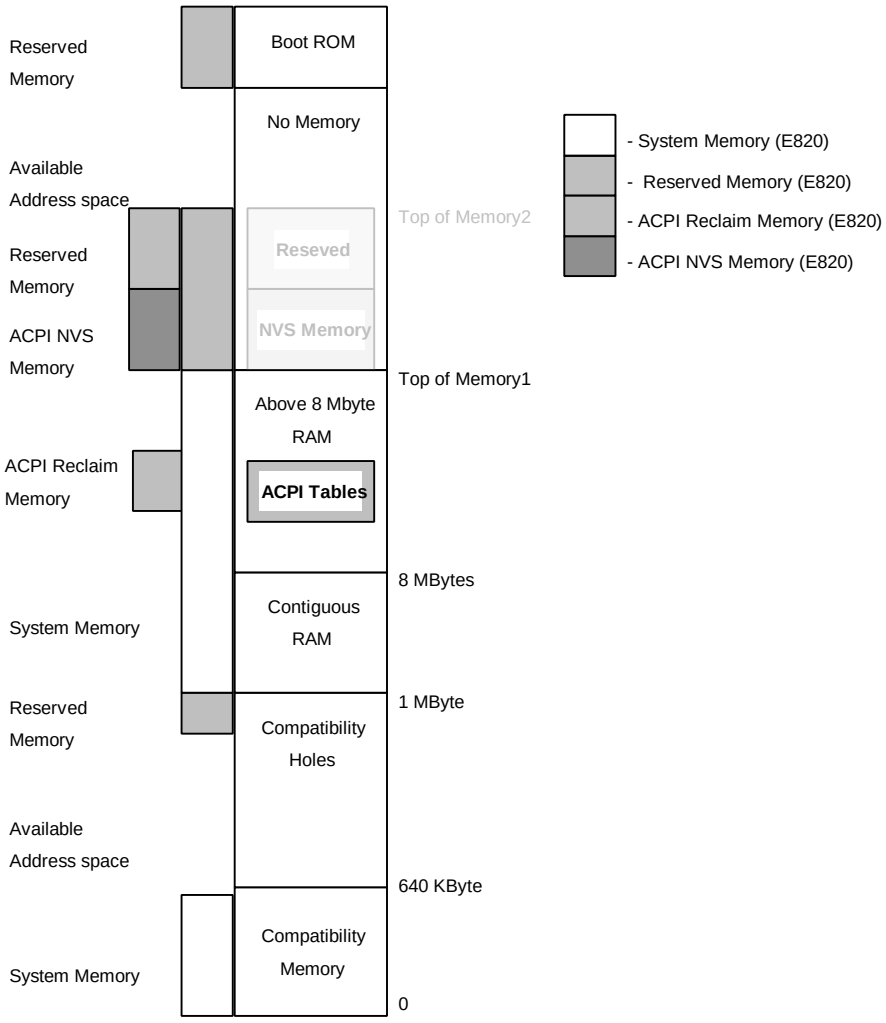

Figure 17. ACPI Memory Configuration

Also, from the Top of Memory1 to the Top of Memory2, the BIOS has set aside some memory for its own use and has marked as reserved both ACPI NVS Memory and Reserved Memory. A legacy OS will throw out the ACPI NVS Memory and correctly mark this as reserved memory (thus preventing this memory range from being allocated to any add-in device).

4.3. Post Modifications - Resume Sequence

This section describes the initialization sequences for an ACPI platform. After a reset or wakeup event from an S2, S3, or S4 sleeping state (as defined by the ACPI sleeping state definitions), the CPU begins execution from its boot vector. At this point, the initialization software has many options, depending on what the hardware platform supports. This section describes at a high level what should be done for these different options. Figure 17 illustrates the flow of the boot strap software.

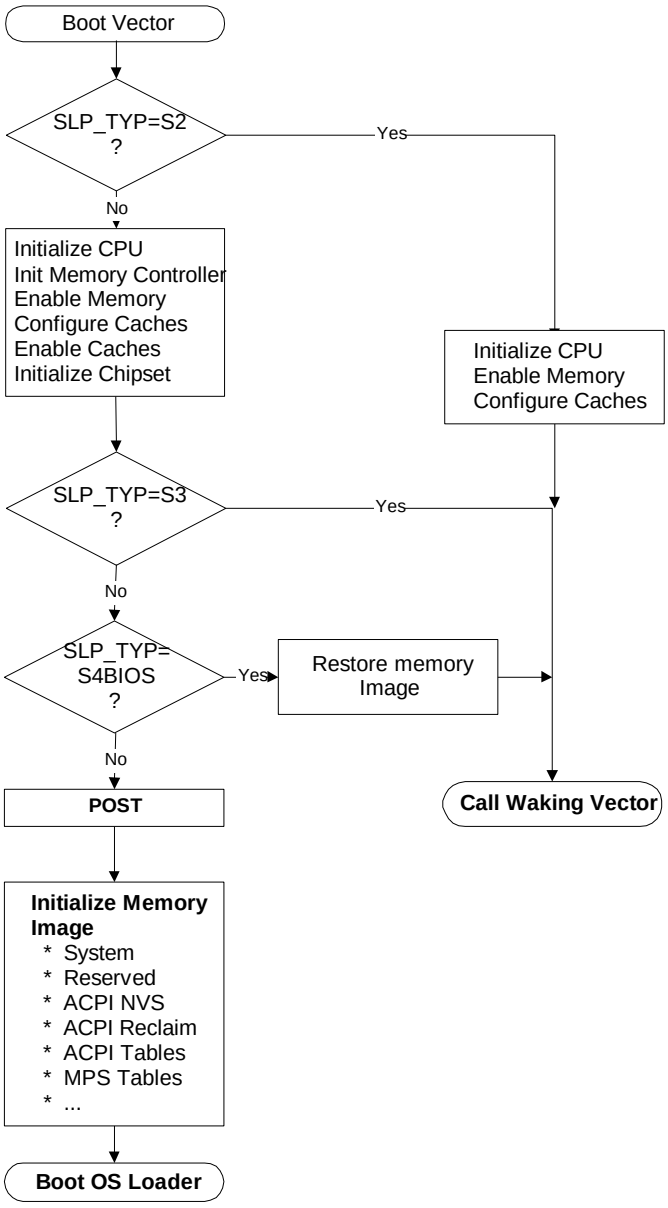

Figure 18. ACPI BIOS Initialization

The processor will begin executing code at its power-on reset vector when either waking from an S2, S3, or S4 sleeping state, responding to a power-on sequence or responding to a hard or soft reset. The sleeping attributes are such that the power-on sequence (and hard and soft reset) are similar to waking up from an S4 state. I.e. the system is configured to a boot configuration and then the OS loader is called. Waking up in the S2, S3, or S4

states only requires a partial configuration by the hardware, followed by calling the waking vector (found in the FACP table).

4.4. OS Loading/Hardware Signature

At this point the BIOS has passed control to the OS, either by using the OS boot loader, (a result of awakening from an S4/S5 or boot condition), or the OS waking vector (a result of awakening from an S2 or S3 state). For the Boot OS Loader path, the OS will get the system memory map through an INT15H E820h call. If the OS is booting from an S4 state, it will then check the NVS image file's hardware signature with the hardware signature within the FACS table (built by BIOS) to determine whether it has changed since entering the sleeping state (indicating that the platforms fundamental hardware configuration has changed during the current sleeping state). If the signature has changed, the OS will not restore the system context and can boot from scratch (from the S4 state). Next, for an S4 wakeup, the OS will check the NVS file to see whether it is valid. If valid, then the OS will load the NVS image into system memory. The OS will then ask BIOS to switch into ACPI mode and will reload the memory image from the NVS file.

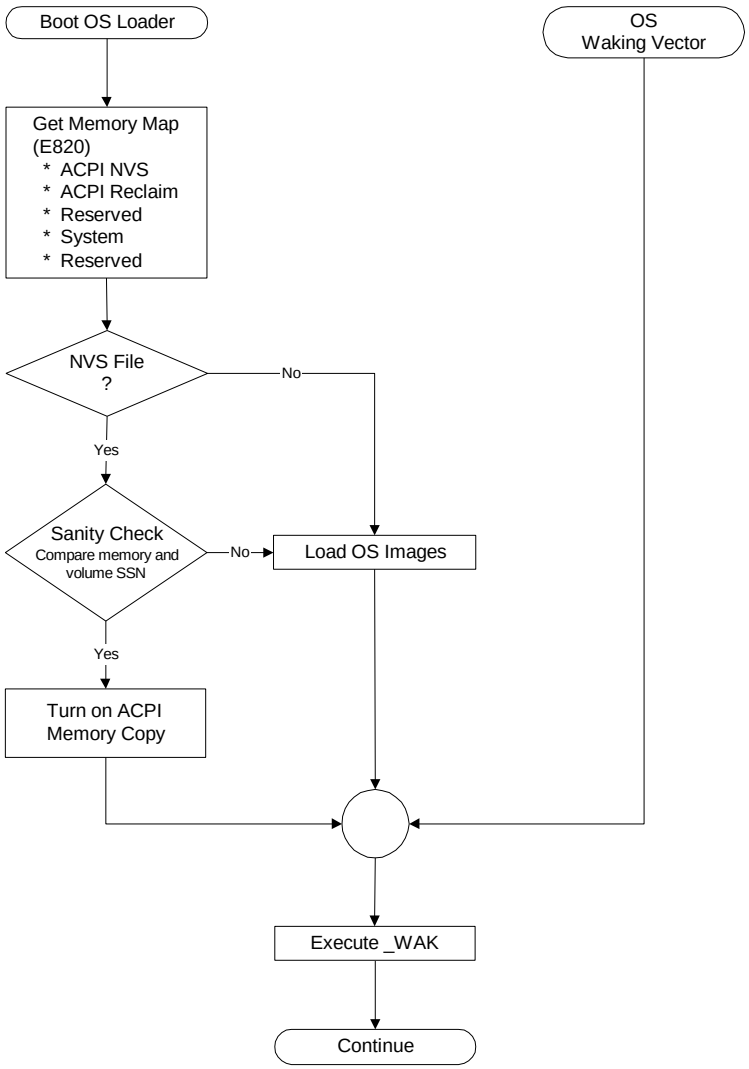

Figure 19. OS Initialization

If an NVS image file did not exist, then the OS loader will load the OS from scratch. At this point, the OS will generate a _WAK call that indicates to the BIOS that its ACPI NVS memory image has been successfully and completely updated.

4.5. Power Management Mode Control

4.5.1. Turning On ACPI

When a platform initializes from a cold boot (mechanical off or from an S4 state), the hardware platform is assumed to be configured in a legacy configuration. From these states, the BIOS software initializes the computer as it would for a legacy operating system. When control is passed to the operating system, the OS will then enable the ACPI mode by first scanning memory for the ACPI tables, and then generates a write of the ACPI_ENABLE value to the SMI_CMD port (as described in the FACP table). The BIOS:

- Saves all Power Management-related chipset registers (such as idle, traps, timers, and so on) in SMRAM
- Disables all legacy PM timers
- Disables all ACPI events so the OS can select the events that match its policies
- Sets the SCI_EN bit in the ACPI hardware register to indicate to the OS that the hardware platform is now configured for ACPI.

When the platform is awakening from an S1, S2 or S3 state, the OS assumes the hardware is already in the ACPI mode and will not issue an ACPI_ENABLE command to the SMI_CMD port.

4.5.2. Turning Off ACPI

ACPI provides a mechanism that allows the operating system to disable ACPI. After unloading all ACPI drivers and disabling all ACPI events, the OS then issues an I/O access to the SMI_CMD port with the value contained in the ACPI_DISABLE field (also in the FACP table). The BIOS:

- Restores the power-management related registers from SMRAM; this puts idles, traps, timers, and so on back to their original states
- Resets the SCI_EN bit in the ACPI hardware register.

If the legacy operating system returns control to the ACPI OS, if the legacy OS has wiped out the ACPI tables (in reserved memory and ACPI NVS memory), then the ACPI OS will reboot the system to allow the BIOS to re-initialize the tables.

5. Dual Mode Power Delivery

The following subsections provide a brief overview of dual mode outputs, a key requirement of the Instantly Available PC. The reader is referred to the Instantly Available PC System Power Delivery Requirements and Recommendations document for a detailed description of the power delivery capabilities that enable the construction of an Instantly Available PC. The first section gives and overview of dual power delivery concept. Next the several implementation options are described including minimum and fully featured capability definition.

Finally the several implementation options are shown with standard ATX power supply, higher standby capacity power supply and fully featured integrated power supply. The section also shows power delivery support for RDRAM memory.

5.1. Concept of dual outputs

A dual mode output is the output of what is essentially a 2:1 mux. The mux is used to route one of its two input voltage sources to a split power plane. Both 3.3V and 5V dual outputs have been defined to support DRAM (active and self-refresh states) and wakeup devices.

The concept of dual output is shown in the following figures.

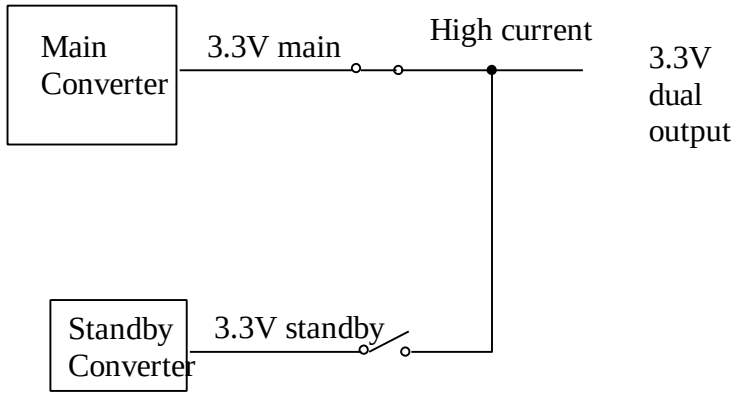

Figure 20. High Capacity Dual Mode Output

When the system is operating normally the output current is drawn from the main power supply converter as shown above. When the system is transitioned from the working state to the sleeping state (e.g. to the Suspend to RAM S3 state), the main converter is turned OFF and the dual output current is then drawn from the standby converter as shown in Figure 21.

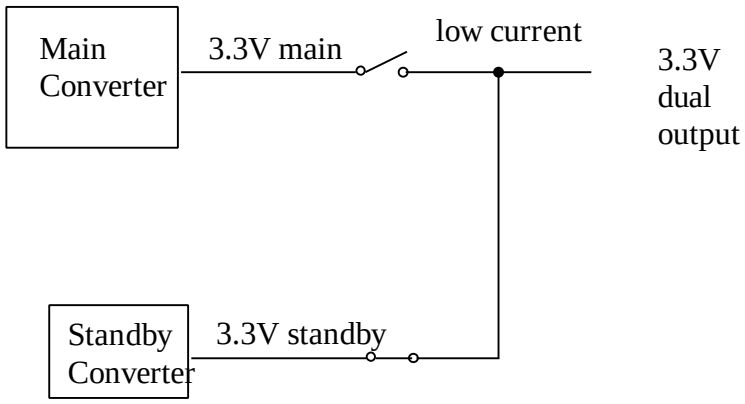

Figure 21. Low Capacity (aux power) Dual Mode Output

Note that Figures 20, and 21 are intended to be conceptual drawings only.

System memory (DRAM) is a typical example of a subsystem that uses the dual mode voltages. During the working state, a typical DIMM module consumes from 3-4 W. However during the sleeping state, the module consumes only between 50-150 mW while in self-refresh mode. All power supply output voltage must remain within the specified regulation band when the system turns off the main converter and switches to the standby converter.

5.2. Minimum DC Output Requirements

The following table defines the minimum set of DC output characteristics required to build an Instantly Available PC. The minimum power delivery subsystem feature set supports:

- Suspend to RAM (ACPI S3 state)
- A single 3.3V PCI add-in or motherboard integrated 3.3V wake device
- Suspend/Resume button
- ACPI controller's resume power well.

Note that in the minimum power delivery subsystem configuration, USB and other 5 volt motherboard integrated wake devices are not supported from the S3 sleeping state (i.e. no support for +5 Vdual).

Output	Standby Mode Min.	Standby Mode Max.	Normal Mode Min.	Normal Mode Green	Normal Mode Max.	Normal Mode Peak
+3.3 Vdc	-	-	0.0 A	1.0 A	9.6 A	-
+3.3 Vdual	0.0 A	700 mA	0.0 A	1.0 A	6.0 A	-
+3.3 Vsby	0.0 A	250 mA	250 mA	250 mA	250 mA	-
+5 Vdc	-	-	0.4 A	1.4 A	16.0 A	-
+5 Vdual	0.0 A	0.0 A	0.0 A	0.0 A	0.0 A	-
+12 Vdc	-	-	0.0 A	0.0 A	1.4 A	4.5 A
-12 Vdc	-	-	0.0 A	0.0 A	0.5 A	-

The following is an itemized breakdown of what the minimum required dual mode output capacity can support while in the S3 sleep state:

	1 - PME# enabled PCI slot		375 mA
	3 - PME# disabled PCI slots		60 mA
	4 - 64MB SDRAM DIMMs		160 mA
			---
		700 -	595 mA
	3.3Vdual headroom ….		105 mA
3.3Vdual	---		
3.3Vsby	ACPI resume well, LEDs, Sus/Res Button		250 mA

5.3. Dual Mode Output Headroom

The following table details a recommended set of DC output characteristics that, when implemented, support a more fully configurable Instantly Available PC. Power delivery subsystems built to the recommended specifications below support:

- Suspend to RAM (ACPI S3 state)
- Multiple I/O bus or motherboard integrated wake devices
- Suspend/Resume button
- The ACPI controller's resume power well.

Note that added support for +5 Vdual enables "off yet communicating capabilities" for USB, and other 5 volt motherboard devices.

Output	Standby Mode Min.	Standby Mode Max.[4]	Normal Mode Min.	Normal Mode Green [1]	Normal Mode Max.[2]	Normal Mode Peak[3]
+3.3 Vdc	-	-	0.0 A	1.0 A	13.0 A	-
+3.3 Vdual	0.0 A	2.0 A	0.0 A	1.0 A	8.0 A	-
+3.3 Vsby	0.0 A	250 mA	250 mA	250 mA	250 mA	-
+5 Vdc	-	-	0.4 A	1.4 A	16.0 A	-
+5 Vdual	0.0 A	1.5 A	0.0 A	0.2 A	4.0 A	-
+12 Vdc	-	-	0.0 A	0.0 A	4.5 A	6.0 A
-12 Vdc	-	-	0.0 A	0.0 A	0.5 A	-

5.4 Power Distribution and Control

5.4.1 Good, better and best power delivery implementations

To allow gradual transitioning of the current power delivery systems to the "Instantly Available PC" power delivery system the following three options have been chosen as the examples with the reference cost targets. The three options refereed as good, better and best are just the implementation examples. More permutations of listed capabilities are also allowed. It is up to the system designer to choose the set of the features for the power delivery subsystem to meet specific system requirements.

Option	3.3V dual output	5V dual output	Supported wakeup devices	Active mode fan control	Efficiency at 3W system load	Cost
Good	Yes	No	1 device 700 mA dual capacity	No Off in S3	min 35%	$1.5-2.00
Better	Yes	No	2 devices 1.2A dual capacity	Yes	min 35%	$3.00
Best	Yes	Yes	4 devices 2A/3.3Vdual 1./A 5V dual	Yes	min 50%	$4.5

Table 9. Good, better, best power delivery options

The detail implementation recommendation for each option is described in the following sections.

5.4.2 Implementation with standard ATX power supply (Good option)

The following section describes implementation using standard ATX power supply with 700mA standby current. Please note that USB wakeup is not supported by this implementation.

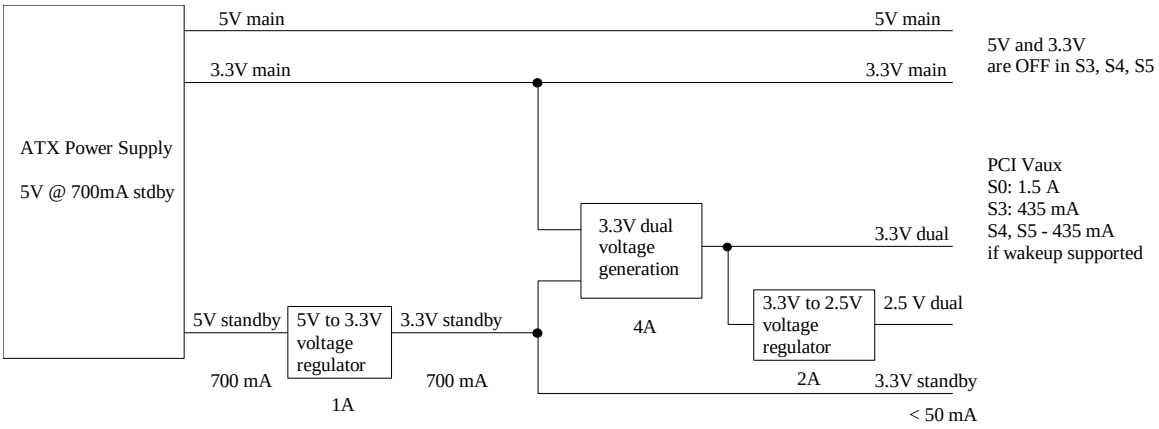

Figure 22. Power delivery subsystem with ATX PS 5V/700 mA standby power

In this implementation the three blocks shown in Figure 22 need to be implemented on the motherboard.

The 5V to 3.3V voltage regulator can be a simple three pin voltage regulator (for example LT1117) with max current capacity of 0.7A-1A.

The regulator to supply the voltage to the RDRAM array has to be rated based on the maximum current requirements for the RDRAM array in the active state.

The dual voltage generation circuit can be implemented using pair of FETs with control logic or other equivalent circuit. The example of dual voltage generation logic is included in the following section. This dual voltage generation circuit can be disabled upon transition to S4 and S5 circuit. (However when the system depends on the wakeup events from either S4 or S5 state the power needs to be retained.)

While in S4 and S5 state the only power that is needed in the system is the power for the ON/OFF switch and the ICH resume well.

5.4.3 Implementation with ATX power supply with 1.2 A standby current (Better option)

This section describes implementation of the “Instantly Available PC” power delivery system utilizing ATX power supply with 5V output and 1.2 A standby current. This is refereed as the “Better” implementation option.

The additional standby current capability is required to support multiple wakeup devices capable of waking up from S3 state. The wakeup devices can be either two PCI devices or PCI and USB device.

Two implementation examples have been described in the next sections. The example 1 shows typical implementation with two circuits to generate dual voltages for 3.3V and 5V. The example 2 optimizes the cost of the power delivery circuit by eliminating 3.3V dual generation circuit.

5.4.3.1 Example 1

The power delivery subsystem can be implemented using two dual voltage generation circuits and two regulators as described in the following picture. The power consumers with estimated power consumption are listed on the right side of the figure. This implementation can support two wakeup devices for example USB and PCI device.

Based on the power budgets only one USB port can be supported for wakeup. If two USB ports and PCI device is enabled for wakeup with only 1.2 A standby current, it is possible that all three devices will generate wakeup events at the same time causing the 5V standby voltage to drop and possibly corrupting system memory.

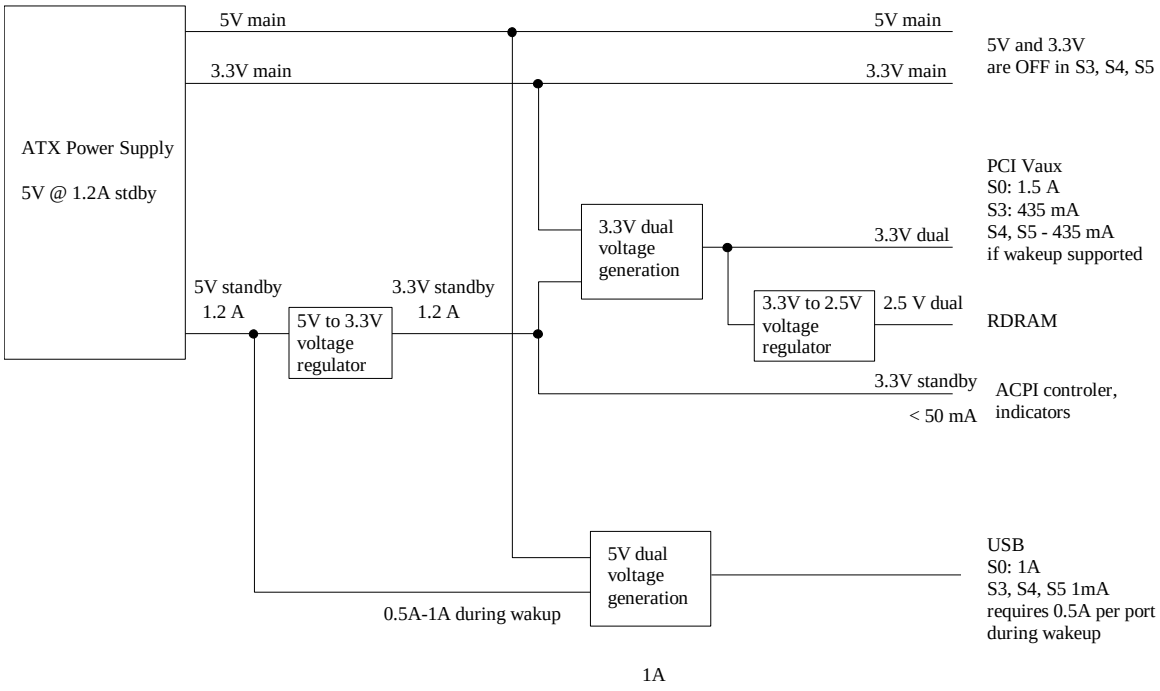

Figure 23. Power delivery subsystem with ATX PS 5V/1.2A standby power – example 1

5.4.3.2 Example 2

The other implementation example is shown in the Figure 24. In this implementation the 3.3V dual voltage generation circuit can be eliminated. The 5V power supply output has to be capable of delivering the appropriate additional power to 3.3 Vaux and RDRAM during normal mode of operation.

The dotted line shown in the Figure 24 illustrates alternate connections in case when USB tolerance can not be met.

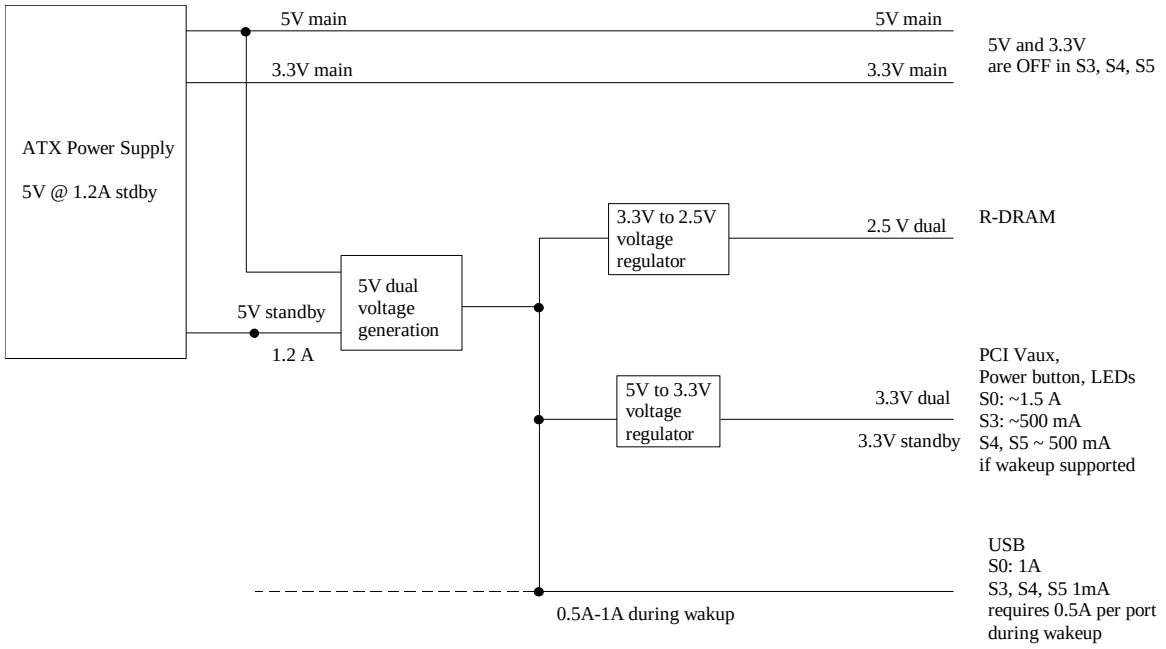

Figure 24. Power delivery subsystem with ATX PS 5V/1.2A standby - example 2

5.4.4 Implementation with integrated power supply as specified in Instantly Available Power delivery recommendations (Best option)

The following Figure 25 describes implementation of the power delivery subsystem using integrated power supply as specified in "Instantly Available PC" power supply specification and recommendations.

This implementation allows four PCI wakeup devices and two USB wakeup devices. Also all the power components have been integrated into the "silver box" power supply. It is expected that the price adder for the fully featured integrated power supply is $4-5. It is also expected that the price adder will be much less when the volume of the supplies will increase.

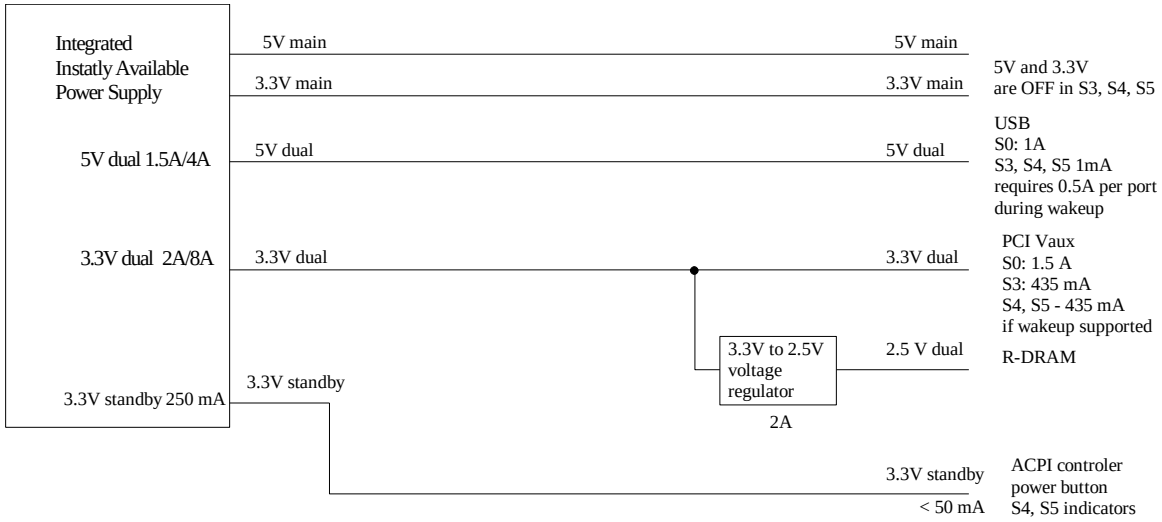

Figure 25. Power delivery subsystem with Integrated Power Supply solution

5.5 Integrated Power Supply overview (a.k.a. PS '98)

The "Instantly Available PC System Power Delivery Requirements and Recommendations" specification describes the features necessary to optimally support the "Instantly Available PC" system.
.

The three most important features in the dual mode supply specification in rank priority are:

1. Integrated support for the S3 (Suspend to RAM) state
2. Increased auxiliary power supply capacity
3. Improved conversion efficiency at light loads

The new power supplies were designed to support Suspend to RAM functionality and also to support 5V and 3.3V wakeup devices. The new power supply is also designed to deliver improved conversion efficient when lightly loaded. This new feature is intended to help the PC meet very low power European energy standards.

5.5.1 New Power Supply Features

The following list outlines the key improvements brought to the PC by integrated power supply:

- Dual mode outputs to support suspend to RAM and PCI auxiliary voltage
  - 3.3V and 5V dual outputs
- 15 W of standby power delivered to the system.
- Improved efficiency for standby mode of operation
  - Targeted at 50% and higher with the AC power ~5W
- 3.3 V always present standby voltage
- Pulse width modulated fan speed control

6. System Software considerations

6.1 BIOS considerations

The BIOS for Instantly Available PC must be ACPI compliant in compliance with ACPI specification rev. 1.0. ACPI hardware confidence tests are available to test system ACPI compliance.

Special BIOS considerations are also described in the PCI PM specification.

6.2 Device drivers

To ensure correct operation of "Instantly Available PC" the device driver is responsible to ensure that all device content is saved upon transition to D3 state and restored when transitioning back to D0 state. When in D3 state the power from the device can be removed without further notification.

6.3 Power Budgeting software

The power capacity of the standby power supply has been chosen to support self-refresh of large memory configuration as well as multiple wakeup devices. In cases where the total power needed for memory and wake up devices exceeds the maximum power specified by the standby power supply, the power management policy manager should not allow the system to enter the low power state which would have the power supply switch over to the standby supply.

However it is strongly recommended that the "Instantly Available PC" be designed with a high priority given to being able to sleep with the main converter off. This is fundamental to enabling the PC to reach a very low power sleeping state where convection cooling is adequate.

6.4 Test Procedures

The following check procedure was used during the Instantly Available plugfest to ensure basic functionality of the systems.

General System Testing

Test #	Test	Comments
ACPI HCT		
	Results of ACPI HCT	Tests ACPI compliance of the BIOS.
PMTest Utility		
	PCI-PM Compliant?	Checks for the proper implementation of PCI PM specification.
Suspend/Resume Testing		
	S1 Test	Functional test, verifies proper transition to S1 state and from S1.
	S3 Test	Functional test, verifies proper transition to S3 state and from S3. Graphics driver save/restore

Application Testing		
	"FuryS1" System sus./res.	More stressful graphics test
	"FuryS1" app. Sus./res.	
	"FuryS3" System sus./res.	
	"FuryS3" app. Sus./res.	
	Additional tests to verify 3D capabilities. 3D Winbench, Tunnel	

Modem/Wake on Ring Testing (Additional Testing for Modem Vendors)

HyperTerminal		
	S1 Wake on Ring	Verifies PCI PM wake capabilities. PME# assertion. Driver and OS interactions
	Connection Established?	
	S3 Wake on Ring	Driver save/restore capabilities.
	Connection Established?	

NIC/Wake on LAN Testing (Additional Testing for NIC Vendors)

Test #	**Test**	**Resume times**
Wake on Ping		
	S1 Wake on Ping	Verifies PCI PM wake capabilities. PME# assertion. Driver and OS interactions
	S3 Wake on Ping	Driver save/restore capabilities.
	S1 Wake on Drive Access	This is more high level test. When to systems are connected to the network the sleeping PC is waken up based on the directory access. The sleeping PC wakes up and services the request.
	S3 Wake on Drive Access	

Audio Resume Testing (Additional Testing for Audio Vendors)

Test #	**Test**	**Resume times**
Media Player		
	Audio after S1 resume.	PCI PM compliance.
	Paused Audio after S1 res.	
	Audio after S3 resume.	PCI PM compliance. Driver save/restore from S3 state.
	Paused Audio after S3 res.	